I

Proceedings of the 9th International Conference on Computational Creativity

Editors: François Pachet • Anna Jordanous • Carlos León

June 25 — 29, 2018 • Salamanca, Spain

C

Association for Computational Creativity

International Conference on Computational Creativity

ICCC'2018

Proceedings of the Ninth International Conference on Computational Creativity

ICCC 2018

Salamanca | 25 - 29 June

François Pachet, Anna Jordanous, Carlos León (Editors)

Published by the Association for Computational Creativity (ACC)

Association for Computational Creativity

University of Salamanca

Spain

http://computationalcreativity.net/iccc2018/

First published 2018

TITLE: Proceedings of the 9th International Conference on Computational Creativity (ICCC'18)

EDITORS: François Pachet, Anna Jordanous, Carlos León

ISBN: 978-989-54160-0-4

Published by the Association for Computational Creativity (ACC)

ISBN 978-989-54-1600-4

9 789895 416004

Program Committee

Senior PC Members

Amílcar Cardoso - University of Coimbra
Pablo Gervás - Universidad Complutense de Madrid
Ashok Goel - Georgia Institute of Technology
Mary Lou Maher - University of North Carolina Charlotte
Alison Pease - University of Dundee
Rafael Perez Y Perez - Universidad Autónoma Metropolitana at Cuajimalpa
Graeme Ritchie - University of Aberdeen
Rob Saunders - Falmouth University
Tony Veale - University College Dublin
Dan Ventura - Brigham Young University
Geraint Wiggins - Queen Mary University of London

PC Members

Margareta Ackerman - Florida State University
Wendy Aguilar - UNAM
Tarek Besold - City, University of London
Debarun Bhattacharjya - IBM
Josep Blat - Universitat Pompeu Fabra
Paul Bodily - Brigham Young Univeristy
David C Brown - CS Dept., WPI
Simon Colton - Falmouth University
Michael Cook - Goldsmiths College, University of London
Joseph Corneli - Department of Computing
João Correia - University of Coimbra
Arne Eigenfeldt - Simon Fraser University
Andrés Gómez de Silva Garza - Instituto Tecnológico Autónomo de México
Hugo Gonçalo Oliveira - University of Coimbra
Jeremy Gow - Goldsmiths, University of London
Kazjon Grace - UNC Charlotte
Sascha Griffiths - University of Hamburg
Christian Guckelsberger - Computational Creativity Group, Goldsmiths College, University of London
Ivan Guerrero - UNAM-IIMAS
Matthew Guzdial - Georgia Institute of Technology
Sarah Harmon - UCSC
Raquel Hervás - Universidad Complutense de Madrid
Amy K Hoover - New Jersey Institute of Technology
Mikhail Jacob - Georgia Institute of Technology
Colin Johnson - University of Kent
Maximos Kaliakatsos-Papakostas - School of Music Studies, Aristotle University of Thessaloniki
Anna Kantosalo - University of Helsinki
Robert Keller - Harvey Mudd College
Oliver Kutz - KRDB Research Centre for Knowledge and Data, Free University of

Bozen-Bolzano
Nada Lavra - Jozef Stefan Institute
Antonios Liapis - Institute of Digital Games, University of Malta
Heather Ligler - Georgia Institute of Technology
Simo Linkola - Helsingin Yliopisto
Maria Teresa Llano - Goldsmiths, University of London
Phil Lopes - Institute of Digital Games, University of Malta
Roisin Loughran - UCD
Penousal Machado - University of Coimbra
Pedro Martins - University of Coimbra
Jon McCormack - Monash University
Stephen Mcgregor - Queen Mary University of London
David Meredith - Aalborg University
Diarmuid O'Donoghue - Maynooth University
Ana-Maria Olteteanu - SFB-TR8 Spatial Cognition, Bremen Universität
Philippe Pasquier - Simon Fraser University
Enric Plaza - IIIA-CSIC
Senja Pollak - Jozef Stefan Institute
Mei Si - Department of Cognitive Science, Rensselaer Polytechnic Institute (RPI)
Adam M Smith - University of California Santa Cruz
Bob Sturm - Queen Mary University of London
Anne Sullivan - University of Central Florida
Tapio Takala - Aalto University
Julian Togelius - New York University
Hannu Toivonen - University of Helsinki
Tatsuo Unemi - Soka University
Frank van der Velde - University of Twente
Lav Varshney - University of Illinois at Urbana-Champaign
Luis Fabricio - Wanderley Goes Pontificia Universidade Catolica de Minas Gerais
Georgios N. Yannakakis - Institute of Digital Games, University of Malta
Martin Znidarsic - Joef Stefan Institute
Oliver Bown - Design Lab, University of Sydney
Daniel Brown - University of Waterloo
Pablo Delatorre - Universidad de Cádiz
Alan Tapscott - Universidad Pompeu Fabra
Susana Bautista - Universidad Francisco de Vitoria

Preface

This volume contains the papers presented at ICCC 2018, the 9th International Conference on Computational Creativity held in Salamanca, Spain from June 25th - June 29th, 2018 http://computationalcreativity.net/iccc2018/. The conference was hosted at the University of Salamanca. ICCC 2018 was the first ICCC organized after the Association for Computational Creativity was formally set up as a legal organisation.

Computational creativity is the art, science, philosophy and engineering of computational systems which, by taking on particular responsibilities, exhibit behaviours that unbiased observers would deem to be creative. As a field of research, this area is thriving, with progress in formalising what it means for software to be creative, along with many exciting and valuable applications of creative software in the sciences, the arts, literature, gaming and elsewhere. The ICCC conference series, organized by the Association for Computational Creativity since 2010, is the only scientific conference that focuses on computational creativity alone and also covers all its aspects.

We received 67 paper submissions, in five categories:

1. Technical papers advancing the state of art in research [papers posing and addressing hypotheses about aspects of creative behaviour in computational systems];
2. System and resource description papers [papers describing the building and deployment of a creative system or resource to produce artefacts of potential cultural value in one or more domains];
3. Study papers presenting enlightening novel perspectives [papers which draw on allied fields such as psychology, philosophy, cognitive science, mathematics, humanities, the arts, and so on; or which appeal to broader areas of Artificial Intelligence and Computer Science in general; or which appeal to studies of the field of Computational Creativity as a whole];
4. Cultural application papers [papers presenting the usage of creative software in a cultural setting];
5. Position papers arguing for an opinion [papers presenting an opinion on some aspect of the culture of Computational Creativity research, including discussions of future directions, past triumphs or mistakes and issues of the day].

Each submission was reviewed by 3 program committee members and then discussed among the reviewers, if needed, to resolve controversial and borderline cases. Senior Program Committee Members led discussions and also prepared recommendations based on the reviews and discussions. In total, around 300 reviews and meta-reviews were carried out in the process. Papers were accepted based on quality, academic rigour and relevance to one or more of the conference's five paper categories.

The committee accepted 38 full papers. Papers were presented either as oral presentations, posters or demos, depending on the nature of the contribution. The three-and-a-half days of the ICCC 2018 scientific program consisted in a series of exciting sessions for oral presentations of papers and a special session for posters and demos.

This conference included a number of satellite events related to creativity and computers, including two workshops, one tutorial and a Doctoral Consortium and an industry panel.

The two workshops were the 6th International Workshop on Musical Metacreation (MUME) and a workshop on Digital Humanities And Computational Creativity (DHCC). The MUME workshop also hosted a concert of musical metacreation. The tutorial organised was “The Shape of Strings to Come”.

ICCC 2018 gave several awards including the Best Paper Award and the Best Student Paper Award.

We thank our sponsor, from which we received very useful support: Spirit AI.

We thank the program committee and the senior program committee for their hard work in reviewing papers and the EasyChair platform that made our work easier. We also thank all those involved in organising ICCC 2018, the ACC steering committee, best paper reviewers and those involved in organising and supporting the workshops, tutorials and doctoral consortium.

ICCC 2018 organising committee

GENERAL CHAIR: François Pachet, Spotify Creator Technology Research Lab, France

PROGRAM CO-CHAIR: Anna Jordanous, University of Kent, UK

PROGRAM CO-CHAIR: Carlos León, Complutense University of Madrid, Spain

LOCAL CO-CHAIR: Juan Manuel Corchado, University of Salamanca, Spain

LOCAL CO-CHAIR: María Navarro, University of Salamanca, Spain

WORKSHOP CHAIR: Sara Rodríguez González, University of Salamanca

INDUSTRY CHAIR: Michael Cook, The Metamakers Institute, UK

PUBLICITY CHAIR: Fernando de la Prieta, University of Salamanca, Spain

SPONSOR CHAIR: Javier Bajo Pérez, Technical University of Madrid, Spain

June 2018

Table of Contents

Co-Creative Conceptual Art

Margareta Ackerman*
Santa Clara University
Santa Clara, CA
mackerman@scu.edu

James Morgan
San Jose State University
San Jose, CA
james.morgan@sjsu.edu

Christopher Cassion
Georgia Institute of Technology
Atlanta, Georgia
ccassion3@gatech.edu

Abstract

This paper explores a conceptual art video presented at Works Gallery San Jose. *Arido Taurajo* is an aria in the style of Giacomo Puccini, set in World of Warcraft. The aria was co-created with Roboccini, a lyrics-to-melody system, made specifically for this artwork. We discuss the human-machine collaboration between Roboccini and the project director, and address both expert and audience reception to the work. We conclude with a discussion on conceptual art and computational creativity.

Introduction

Unlike the plastic arts[1], conceptual art is at its core atypical, de-emphasizing the object and aesthetics, and focusing instead on the meaning behind the artwork. Conceptual art frequently reflects on meaning through the process of fabrication rather than representation of the finished art object. This idea was introduced to the arts by Marcel Duchamp with the submission and subsequent rejection of Fountain to the Society of Independent Artists in 1917. Duchamp liberated art from media and demanded that it be about ideas rather than aesthetics.

Conceptual art often appears at the intersection of art & technology, like New Media, or reflects on arts inspiration, like Post Internet Art, but it always looks over its shoulder at concept or meaning of the artwork. *Arido Taurajo* combines a new approach to music creation with machinima video to tell an old and yet familiar story about life and family in a modern context.

Arido Taurajo is the first aria composed by a non-musician. This was made possible by making a co-creative system that enables human collaborators (who may not have musical expertise) to write melodies for their lyrics. To this end, we made Roboccini, which creates melodies in the style of the famous opera composer Giacomo Puccini. Roboccini takes in Italian lyrics, and responds with melodies to which the lyrics can be sung. It consists of machine-learning models trained on Puccini's music, as well as incorporating co-creative functionality to give the human user greater creative control.

Arido Taurajo's director, James Morgan, collaborated with Roboccini to create the entire melody for the aria. The complete artwork relied on additional human collaborators, including a singer, producer, and machinimist, resulting in an exhibition in the 40th Anniversary show *Making it Works* at the Works Gallery, San Jose (Figure 1). Video excerpt found here `https://www.youtube.com/watch?v=6G_LmxWYUOU`.

In this paper, we discuss the nature of the co-creative process used to create the music for *Arido Taurajo*. In particular, we address how the director's experience co-creating with Roboccini contrasts with prior experience collaborating with human musicians. Working with Roboccini was more satisfying, and brought James deeper into the creative process. Unexpectedly, Roboccini ended up teaching through extended experience. James began to understand musical notation, and complexity of the melodies with respect to the singer's range and ability.

Arido Taurajo showcased at the 40th anniversary exhibition "Making It Works" Works Gallery and at the Paseo Prototyping Festival, San Jose, elicited the reaction of art experts. We share and discuss expert comments, as well as those of lay audiences, spanning comment on the artwork as a whole, as well as specifically addressing the reception of art co-creation by a machine.

The introduction of Computational Creativity into the process, hereafter referred to as CC, and into the conceptual art world gives rise to important questions. Conceptual art challenges typicality, and focuses instead on meaning. The current work demonstrates the participation of a machine collaborator in the role of an expert - that is, a collaborator with a well-defined expertise, who doesn't need to focus on the global objectives of the resulting artwork. We discuss how the current work, as well as art co-created by a machine in general, fit within the framework and history of Conceptual Art.

The paper begins with a discussion and technical description of Roboccini. Next, we report and discuss expert and audience reception, followed by a detailed account of the co-creative process underlying the creation of *Arido Taurajo*. We conclude with a discussion of creativity, conceptual art, and the future of large-scale productions made in collabora-

*The development of Roboccini was conducted while the first author was at San Jose State University.

[1]The plastic arts refers to physical media, painting, sculpture, photography, film etc. and as a description has come to refer to all visual arts.

Figure 1: Installation View *Arido Taurajo*, Leaving Orgrimmar, 40th anniversary exhibition "Making It Works" Works Gallery San Jose, image: Joe Miller

tion between humans and machines.

Roboccini

Roboccini is a co-creative melody writing system which takes in lyrics in Italian and in returns melodies for these lyrics. Trained on the music of the famous Italian composer Giacomo Puccini, this system was made with the aim of giving *Arido Taurajo* Puccini's recognizable, grand style.

This section begins with a brief discussion of related previous work, after which we present the models underlying Roboccini, and discuss its co-creative features.

Related previous work

Compared with the wealth of systems for creating lyric-free music (See, for example, an excellent overview by (Fernández and Vico 2013)), algorithmic songwriting is still in its infancy. Several songwriting systems explore the potential of autonomous algorithmic songwriting. For example, M.U. Sicus-Apparatus (Toivanen, Toivonen, and Valitutti 2013) demonstrates how the entire songwriting process can be integrated, from lyric generation to melody and accompaniment. Another interesting system, SMUG (Scirea et al. 2015), autonomously creates songs using lyrics based on academic papers. See (Ackerman and Loker 2017) for a more detailed exposition of previous work on algorithmic songwriting.

Interaction between human and machine in musical creation primary focused on music without lyrics. For instance, interactive evolutionary computation typically allows the user to take on the role of a fitness function (Onisawa, Takizawa, and Unehara 2000), (Takagi 2001), (Johanson and Poli 1998), (Collins 2002). Another related line of work studies human-computer improvisation (Keller et al. 2006), (Kitani and Koike 2010).

A recent lyrics-to-melody writing system, ALYSIA (Ackerman and Loker 2017), was used to create English pop songs, and later applied towards the creation of art songs for Emily Dickinson's poetry (Cassion et al. 2017). Songs created with ALYSIA can be found at `http://mayaackerman.info/alysia.html`. ALYSIA aims to ease the songwriting process for the human user without impeding creative self-expression, in a manner useful for both professional musicians, and amateurs who may not be able to engage in this artform without the aid of a songwriting machine.

The current system, Roboccini, challenges ALYSIA's limits by training it on songs that are different from ALYSIA's original training data in both musical style and language. Unsurprisingly, there are stark differences in the music resulting from interaction with Roboccini versus that made with ALYSIA.

Roboccini was the first to incorporate co-creative functionality, which gives the user greater control over the melodies created by the system. For example, we've allowed the user to ask Roboccini to generated similar melodies to a generation that it has previously provided. Co-creative functionality was later also incorporated back into ALYSIA (Cassion et al. 2017).

Figure 2: *Arido Taurajo*, Dahlia flying over the Barrens on her way home to her family. Image James Morgan, video Chantal Harvey

Models and Features

Roboccini is based on a random forest model, trained on a corpus of Giacomo Puccini operas. The data was first gathered, pruned, then split into a training (75%) and test set (25%). For model building and evaluation, the data is split using stratified sampling along the outcome variable, which is scale degree with accent for the melody model, and note duration for the rhythm model.

The training data is built from a set of 25 MusicXML files containing complete Puccini arias. For each of the files, we identified the lead parts and extracted both lyrical and musical features. This led to about 3500 observations. Each observation consists of a combination of lyrical and musical features for the current note as well as the five previous notes. With these observations, we built two predictive models: one for pitch, and the other for rhythm/duration. Roboccini's melody model benefits from the output of the rhythm model which causes it to incorporate the latter's probability.

One of the main differences between ALYSIA and Roboccini is the language switch from English to Italian. Stress testing the structure and process of the base system using a different language and musical style proved challenging due to the required lyrics features and the increase in the complexity of song structure. Unlike English, there isn't as much support for natural language processing for the Italian language. To accurately extract the appropriate features to train the models, we had to adjust how we calculate our text-based feature metrics such as word frequency and number of syllables. We made use of a large freely available corpus of Italian works as a substitution to the Brown corpus (Baroni and Bernardini 2006).

Features extracted include:

- First Measure - A boolean variable indicating whether or not the current note belongs to the first measure of the piece
- Key Signature - Key signature of the current note
- Time Signature - Time signature of the current note
- Offset - The number of beats since the start of the music
- Offset within Measure - The number of beats since the start of the measure
- Duration - The length of the note
- Scale Degree - The scale degree of the note (1-7)
- Accidental - The accidental of the note (flat, sharp, none)
- Offbeat - A boolean variable specifying whether or not the note is offbeat
- Syllable Type - Classifies the current syllable in one of the following four categories: Single (the word consists of a single syllable), Begin (the current syllable is the first one in its word), Middle (the current syllable occurs in the middle of its word), End (the current syllable is the last one in its word).
- Syllable Number - The syllable number within its word
- Word Frequency - The word frequency of the word which the current note/syllable is part of. This value is obtained through the dictionary frequency obtained by indexing the words of a large Italian works corpus.
- Word Rarity - A function of word frequency, as defined by (Nichols 2009). $WordRarity = 2(1 - \frac{\log_{10}(WordFrequency)}{7})$.
- Scale Degree, Accidental, and Duration of previous 5 notes

Co-Creative Functionality

As mentioned above, Roboccini was the first to incorporate co-creative features to facilitate the co-creative process.

Similarity by Distance Opera, being a sung story, can contain repeated motifs. The aim of this feature is to allow the user to repeat certain motifs in their composition. This functionality extracts note information of a user-chosen phrase and performs the cosine distance between the newly generated option and the user-chosen phrase. With this feature, the user specifies a generation and Roboccini orders newly generated melodies from closest to farthest, allowing them to discover similar melodies to those they like. This not only allows for repeated musical ideas but also helps maintain the general structure of the piece.

Connecting Phrases In order to ease the process of connecting newly generated phrases with old ones, this feature allows the user to pick any previous phrases, and create new ones that are of varying degrees of similarity. Since each of the phrases are generated independently from each other, this functionality allows the Roboccini to generate phrases that connect effortlessly. The way we achieve this is by inputting the last n notes of a sequence, $n \leq 5$, as the previous notes at the start of a new generation. This information allows the first few notes of the new melody to be based on a previous melody. One can think of the new melody as a pseudo-continuation of the former. To allow greater control over the degree of similarity, n is user-specified. This functionality has worked so well with Roboccini that we later added it to ALYSIA (Cassion et al. 2017).

Melody Creation

After the models are trained, they are used to create melodies for user-provided lyrics. Lyrics are given to the system one line at a time. Subsequently, lyric features are generated. For each line, we read the feature set from the lyrics and generate the rhythm followed by the pitches. Within each mode, *we generate one note at a time for each syllable in the lyrical phrases*. Finally, the models are used to generate the melodies, which are returned to the user, where the number of melodies is specified by the users.

In addition to the number of melodies, the user can also specify the time and key signature, as well as a special explore/exploit parameter. When predicting rhythm or scale degree (for each note), the models output a distribution over all possible outcomes. The explore/exploit parameter dictates how many samples, with replacements, we make from this distribution. The final choice is the most common draw, with ties broken by the original outcome distribution. A higher explore/exploit parameter value means we rely on the model more heavily.

Arido Taurajo

Arido Taurajo is a video comprised of an Italian aria, created in collaboration with Roboccini, and World of Warcraft machinima. "Arido Taurajo" loosely translates into "Barren Taurajo" or "Taurajo in the Barrens." Both lyrics and visuals reflect the underlying storyline. Dahlia (our hero) is a Tauren (half bovine minotaur-esque creature) and the region is named after her people. In this first aria, Dahlia is finishing a day of doing repetitive quests ("grinding") at the capitol city of Orgrimmars flight point. She plans to go to her home, have dinner with her husband and tuck her child in before she joins her guild for some high level play ("raiding"). Dahlia respects the people around her, even the Non-Player Characters (NPCs) like Doras the Wind Rider Master. The aria opens with a polite verbal acknowledgement of this NPC. This is a transposition of real life into the World of Warcraft and doesn't make much sense in the context of the game. However, it is potentially of interest to the person playing.

Connecting the world, which is often devoid of humanity, with genuine human urges, creates a beautiful contrast and allows the observer to reflect on what makes us human. Adding the subtle and casual points of contact with the people in the world along with the focus of many of our lives (family) breathes life and passion into an otherwise staid and grindy world. This humanizing factor combined with the nostalgia many players have for the world is part of the message of "*Arido Taurajo*."

A female hero speaks to the challenges of woman gamers and the challenges that women face often being forced to choose between career and family. Our hero, Dahlia, manages to have both, and though she is just returning to the "working world" she is confident and welcomed by her guild.

The central idea behind *Arido Taurajo* is to treat the characters in the game space as though they were real people and to permit them to pursue desires that are uniquely human. The goal of this is to question both reality and the game space. Dahlia's trip home is necessary, but having a family in World of Warcraft is neither possible nor practical (Figure 6).

This is not the final form of the work. Plans include submission to film festivals and gallery exhibitions. We intend to expand this into an operatic short, continuing to use machinima, and eventually to create a fully staged production. The slow time-line is to accommodate the creation of processes along the way and to provide time to iterate on essential features.

Machinima

Machinima, a portmanteau of machine and cinema, uses the computer screen as source material for film production, in our case we are looking at the original World of Warcraft (WoW) (Figure 2). Our reasoning for using this setting is one of practicalities, the space has an epic and nostalgic feeling for many and provides a platform for talking about the human condition. It is "large enough" to support an operatic story. Many games, like WoW, create engagement loops that require players to spend excessive amounts of time in an area doing repetitive tasks ("farming or "grinding"). This activity, while often boring, creates opportunities within the landscape for players to interact and commiserate. This is the back story of the Barrens where *Arido Taurajo* is set.

Expert and Audience Reception

Figure 3: Co-creator of Roboccini and singer, Maya Ackerman, in gallery with *Arido Taurajo*, image James Morgan

"Works is very proud to have presented the gallery premier of Arido Taurajo *as a feature of our 40th anniversary exhibition 'Making It Works.' In this music video production, James Morgan and his collaborators brought together a surprising and inspiring mashup of opera and gaming to the intrigue of audiences from youth to seniors and in between."* - Joe Miller, Board President Works SJ. 3

Arido Taurajo's presence in a modern gallery setting represents a mashup of cultures. Combining gaming culture and opera in a way that appeals to a wider audience.

Several experts and audience members commented on the human-machine collaboration involved in the creative process behind *Arido Taurajo.*

"The merging of humans and technology is inevitable and in-progress. The Roboccini work is a clear and positive example of this marriage. It gives the sensation of human and artificial intelligence augmenting each other as emotion flows through the beautiful melody and voice in collaboration with the computational powers of the AI." - Nina Colosi, Founder Streaming Museum.

Some observers quickly arrived at challenging questions that ponder what it means to collaborate with a machine when they are so profoundly different from ourselves (or are they?).

"Artists have always stretched themselves with new media, whether they worked in solitude or in collaboration. Now, the human artist has not only a new medium but a new collaborator as well. This is taking art to the next logical step in our evolution." - Kelly Harrison, audience member.

Others expressed curiosity about human-machine co-creativity, wishing to dig deeper into the process and understand its flaws, exhibiting profound insight into the challenging nature of new forms of collaboration.

"Some of the most beautiful artworks have derived from collaboration. Often this connection can be tense, challenging but often illuminating. Most importantly collaboration embraces flaws. I wonder where and how the flaws in this AI collaboration affected the outcome of this piece." - Danielle Siembieda-Gribben, artist & audience member.

The piece evoked feedback from musicians, who validated that the project's succeeds in co-creating a classical aria.

"Wow! An app did that! Thats pretty amazing. The melodic line is quite lovely and has that classic Italian aria feel to it." -Stefanie Posner, Music and Education Director

Yet others commented on the work of art as a whole, focusing on the meaning behind the conceptual art piece, and the feelings that it evokes.

"Wow! ... Highly highly impressive! I truly enjoyed having both the visuals and the music together... The tone of the overall composition left me feeling empowered and it also gave me the feeling I get when I listen to Philip Glass's music; where it is a silent empowerment... I also liked that it was sung by a woman (beautiful job Maya!), that again gave me a boost of that empowerment, as if to say that the 'woman is independently traveling this world and has the ability to not only be fearless doing this, but can also control a creature who obeys her commands for travel.' I also enjoyed that I couldn't understand the language sung: that way it allows the viewer to 'choose their own adventure' and use their imagination to picture what she might be saying. Allowing the listener to feel empowered again." - Jennifer Petush, dance/digital choreographer

Arido Taurajo was also shown at the Paseo Prototyping Festival in San Jose along with a discussion of Roboccini and has been shown at the LASER (Leonardo Art Science Evening Rendezvous) series, and in Georgia at MUME - International Workshop on Musical Meta-creation.[2].

In a conference setting *Arido Taurajo* is presented as a video and the audience response continues to amuse the authors. The audience is presented with a cartoony animation (from a game) that quickly changes into a musical number, in the style of Puccini. Audiences are often unsure how to respond, but the humor (Figure 5) eventually overwhelms them and they begin to laugh. The novelty then transfixes the audience for the duration of the song, which is then followed by genuine applause.

"This work renews my faith and interest in digital media art. I didnt think there was anything this creative going on." - comment made to board president at Works SJ exhibition.

The response has been terrific, kids love the game footage and the music; they tolerate that they do not understand Italian. Adults hear the aria and may recognize the style of Puccini, and so they are frequently able to make a connection to the game environment.

As an initial offering for a New Media Art piece, the video and aria function together and the aria can be performed separately or in conjunction with game footage (as happened at MUME 2017 in Atlanta).

Collaboration

James is a conceptual artist who works in the media of games and video (machinima). His work often considers aspects of humanity that follow the avatar into a game, and how the actions within a game environment effect the lives of actual people. James' major works are collaborative, this is related to scale and scope of the work as much

[2] http://musicalmetacreation.org/workshops/mume-2017/call-for-participation/

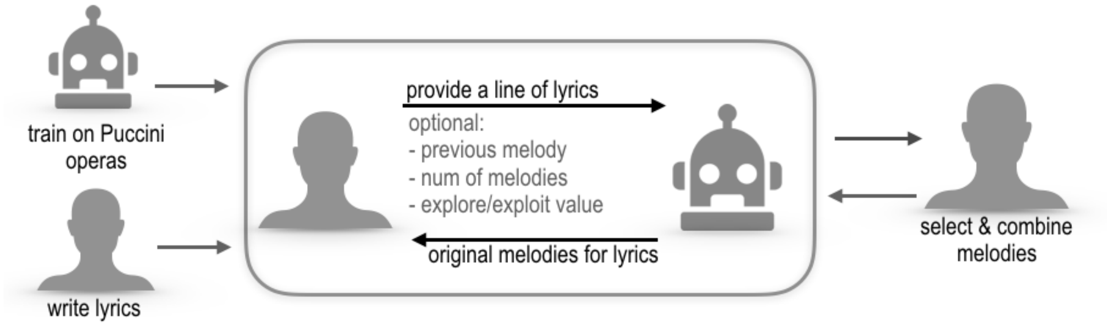


Figure 4: An illustration of the human-computer collaboration in the creation of Arido Taurajo. After training Roboccini on the music of Giacomo Puccini, James repeatedly called Roboccini to generate melodies for one line of lyrics at a time. James then selected amongst Roboccini's suggestions to create the aria.

as his desire to interact with people in the process of creation. Larger works requiring broader expertise demand either greater time or expert collaborators.

As a person with no musical skills, the contrast between collaboration with Roboccini and human musicians is stark. James' earlier work "Einstein's Special Theory of Relativity: The Musical" [3] was written with human collaborators. Human musicians perform like a black box, taking input and creating based on it, the process relies on the aesthetic, passion and commitment of the musicians. Roboccini on the other hand demands that the artist select, arrange and tune. Roboccini opens the process to deep and meaningful collaboration on a granular level. Furthermore, working with Roboccini created learning opportunities. This again becomes part of the collaborative experience. The artist inadvertently started to learn the basics of reading music, and began to understand its basic structure.

Roboccini is the expert and the musician. James had worked with musicians before, but not quite as closely as he had worked with Roboccini. The artist had to feed Roboccini lyrics and sift through its suggestions (see Figure 4 for an illustration of the co-creative process). In the past, working with human collaborators he would present the lyrics, wait a few weeks and have a "finished" work. As an artist, and a director, this is efficient but not terribly satisfying.

In the production of this aria, Roboccini fulfilled the role of an expert collaborator. Roboccini's expertise was baked into every interaction, and never wavered. Initially there were challenges with communication and patterns of production but experience dictates one always adapts to one's collaborators. In this case there was a play between use of the tools and the process for creating the melody for the aria. James had to figure out how to collaborate and frequently asked for revisions, Roboccini always responded. This let him have more of a hands-on approach in a field in which he has no expertise. As director, James had to accommodate the singer, her range and the complexity of the melody.

[3]James Morgan, 2005. Einstein's Special Theory of Relativity: The Musical. IMDB: `http://www.imdb.com/title/tt1557564/`

Roboccini could create a ridiculous amount of melodies and as a collaborator James' task turned to working to refine and simplify these.

Figure 5: *Arido Taurajo* Dahlia sings about a familiar character in Barrens chat, image James Morgan, video Chantal Harvey

Dealing with Roboccini's quirks (only understands 19th century Italian) became a constraint that challenged James' artistic practice. Roboccini understands Italian and James does not, so English lyrics needed to be translated into contemporary spoken Italian before the software was able to generate melodies. Similarly, as is often the case with NLP dictionaries, Roboccini's Italian dictionary was incomplete. This makes it difficult to create meter and rhyme schemes for words outside of Roboccini's scope. Thankfully Roboccini accommodates revisions, and in the end the language difference provides a layer of mystery and style. Viewers recognize Italian, and make a connection with classical opera which gives the work a sense of high culture.

Roboccini also demanded that James give it lyrics one line at a time, but he found that he could cheat by queuing up the lines that he was interested in and having melodies generated for a large volume of content at one time. Using a command line interface made it simple to generate a wide variety of input. This actually became a problem as creating melodies was far too easy.

Roboccini fed his desire to make music and provided some structural constraints, but no constraints on how far he could take it. Roboccini let James retain all of its work and peek behind the creative curtains to see all of its variations, he held onto these during the process of writing and found that it not only gave him readily available alternatives but an almost infinite variety of combinations. This also became useful as he had to accommodate the vocal range of the singer.

Roboccini was limited in one crucial aspect upon which we built an aria and a video. Roboccini understands melodies and Puccini's style, but understands less about the greater meaning of the work. Working with experts demands the ability to look at the project holistically. This responsibility falls on one individual, the director, while Roboccini plays its part as a composer.

Alternating and task-divided human-computer co-creativity is handled organically, with each party taking on functions within its expertise. In this case, the human "director" is responsible for being aware of all the requirements of the collaboration. For example, the range of the singer, the length of the piece, and the "landscape" of the melody. In that way, certain portions of the song can be emphasized and coordinated with the video.

The director relies heavily on the musical expertise of the computer collaborator that is refreshingly eager to redo, replace and re-create various aspects of the melody. Roboccini, in this case is exclusively responsible for certain aspects and is effectively coached into an effective integration of its work.

Discussion: Creativity & Conceptual Art

Arido Taurajo is a work of conceptual art that is by necessity collaborative. It benefits from collaboration because of the number of moving parts and essentially requires expertise in the following areas: environment/game knowledge, libretto, contemporary Italian, melody creation, singing, musical accompaniment and production, puppeteering (machinima), video editing. The director's job is to hold all of this within a single vision and marshall the co-creative process throughout.

How is this a work of conceptual art? Conceptual art breaks from the foundations of art in 1917 with Duchamp's *Fountain*. A urinal placed into an unjuried exhibition (and rejected). This act of artistic creation is acknowledged later by Kosuth as one of the first examples of conceptual work.

Conceptual art is about the idea or meaning behind the art rather than the aesthetics or what it looks like. Kosuth says, "All art (after Duchamp) is conceptual (in nature) because art only exists conceptually." (Smith 2011) For artists, this is liberation from any requirement of form, function, aesthetic or interpretation. Fundamentally due to the movement of conceptual art in the 1960's and 1970's there is no limitation to media or representation, this also frees the "artist" to be non-human.

The tradition of "found art" exemplifies this, "found art" is not art that is created, but art that is recognized by the artist and later described as and "made" into art through context. The context may be a museum or gallery or simply writing or discussing the piece as a work of art.

The current work explores a machine collaborator in the role of an expert, focused on a specific, well-defined task, which the human director utilizes to create the grand vision. The next step may involve machine collaboration on the meaning and global direction in the creation of a piece of Conceptual Art. What would it be like for a conceptual artist to collaborate with a machine that is able to infer and assign meaning, or have creative disagreements with the artist? Works towards creative internationality in CC system applied to the visual arts is an ongoing area of research ((Norton, Heath, and Ventura 2013), (Colton et al. 2015)). Conceptual art, even more so than other art forms, emphasizes meaning and intentionality. Applying machine internationality to collaborative conceptual art may lead both artists and CC researchers to exciting new terrains.

Figure 6: *Arido Taurajo* Dahlia arrives home and is greeted by her husband Diesel and child Calvin, image James Morgan, video Chantal Harvey

Musicals and Operas: The Next Step

Perhaps the most culturally significant large-scale human-machine collaboration was a musical put on at Londons West End for a two week run at the end of 2016. "Beyond the Fence" explored the potential of computational creativity systems to assist with the making of full-scale stage productions (Jordanous 2016). This massive undertaking, which involved both a production and documentary of the creative process behind it (Productions 2016), successfully raised the profile of computational creativity research in the public eye (Jordanous 2016).

Members of the human creative team demonstrated commitment to relying on the output of CC systems, even when it proved challenging, saying "we have to honour what we've signed up for." One of the biggest challenges turned out to be the juxtaposition of text with melody, which stands at the heart of songwriting (Productions 2016). Expert humans were left struggling to create coherent songs that would incorporate parts of music and lyrics created by computer systems.

Naturally, and without downplaying the incredible accomplishment that is "Beyond the Fence," it is safe to say that we would like CC systems to enrich and simply the creative process, rather than pose a challenge.[4] Furthermore, machine collaborators should increase access to creative tasks, enabling the creation of musical, or opera (etc), by humans who may not have been able to participate in this artform without the machine collaborators.

Roboccini, and its sister system, ALYSIA, make a specific creative task, songwriting, both easier and more accessible. Notably, Roboccini solves precisely one of the issues found to be most challenging during the creation of the musical, the juxtaposition between lyrics and melodies.

Roboccini was essential to the creation of *Arido Taurajo*. The director using Roboccini, a non-musician, could not have made the aria without collaborating with Roboccini. Perhaps if another (co-)machine-made musical were to be made today, Roboccini or ALYSIA could make it easier.

What does the future of culture and CC hold? Can we imagine the creation of a musical, or opera, with an effective team comprised of both humans and machines? Rather than struggle to incorporate a machine's suggestions, human artists of the future may be eager to work with machine collaborators that simplify the creative process and fill in skill sets that the humans lack. Machines might empower the human artist, inviting them into the creative process, making the process more transparent than a typically human-to-human collaboration, as Roboccini did for the director of *Arido Taurajo*.

New Media artists hunger for collaborators and new forms of production to play with. The computational creativity community could benefit from early partnerships with artists collaborating with their AI's to co-create new work. Earlier communication in the process of creating and working with AI's will serve to create deeper collaborative connections. Artists, AI's, and CC researchers iterating to better understand each other's requirements may lead to richer and more satisfying collaborations, to the extent that human-machine collaborations in the art world may become commonplace.

References

[Ackerman and Loker 2017] Ackerman, M., and Loker, D. 2017. Algorithmic songwriting with alysia. *EvoMUSART*.

[Baroni and Bernardini 2006] Baroni, M., and Bernardini, S. 2006. *Wacky! Working papers on the Web as Corpus*. Bologna: GEDIT.

[Cassion et al. 2017] Cassion, C.; Ackerman, M.; Loker, D.; and Palkki, J. 2017. With bolts of melody! songwriting with alysia & emily dickinson. In *Musical Meta-Creation (MuMe) Workshop*.

[Collins 2002] Collins, N. 2002. Interactive evolution of breakbeat cut sequences. *Proceedings of Cybersonica, Institute of Contemporary Arts, London, England*.

[4]It is worth noting that not all of the CC systems used in the making of "Beyond the Fence" posed challenges. It is further unsurprising that the very first attempt of a machine-human collaboration of this magnitude would uncover opportunities for improvement.

[Colton et al. 2015] Colton, S.; Halskov, J.; Ventura, D.; Gouldstone, I.; Cook, M.; and Ferrer, B. P. 2015. The painting fool sees! new projects with the automated painter. In *ICCC*, 189–196.

[Fernández and Vico 2013] Fernández, J. D., and Vico, F. 2013. Ai methods in algorithmic composition: A comprehensive survey. *Journal of Artificial Intelligence Research* 513–582.

[Johanson and Poli 1998] Johanson, B., and Poli, R. 1998. *GP-music: An interactive genetic programming system for music generation with automated fitness raters*. University of Birmingham, Cognitive Science Research Centre.

[Jordanous 2016] Jordanous, A. 2016. Has computational creativity successfully made it beyond the fencein musical theatre? In *Proceedings of the 7th International Conference on Computational Creativity*.

[Keller et al. 2006] Keller, R. M.; Morrison, D.; Jones, S.; Thom, B.; and Wolin, A. 2006. A computational framework enhancing jazz creativity. In *Proceedings of the Third Workshop on Computational Creativity, ECAI*.

[Kitani and Koike 2010] Kitani, K. M., and Koike, H. 2010. Improvgenerator: Online grammatical induction for on-the-fly improvisation accompaniment. In *NIME*, 469–472.

[Nichols 2009] Nichols, E. 2009. *Lyric-based rhythm suggestion*. International Computer Music Conference (ICMC).

[Norton, Heath, and Ventura 2013] Norton, D.; Heath, D.; and Ventura, D. 2013. Finding creativity in an artificial artist. *The Journal of Creative Behavior* 47(2):106–124.

[Onisawa, Takizawa, and Unehara 2000] Onisawa, T.; Takizawa, W.; and Unehara, M. 2000. Composition of melody reflecting user's feeling. In *Industrial Electronics Society, 2000. IECON 2000. 26th Annual Conference of the IEEE*, volume 3, 1620–1625. IEEE.

[Productions 2016] Productions, W. 2016. Computer says show. *Two-part UK television documentary, aired on the Sky Arts channel* 25.

[Scirea et al. 2015] Scirea, M.; Barros, G. A.; Shaker, N.; and Togelius, J. 2015. Smug: Scientific music generator. In *Proceedings of the Sixth International Conference on Computational Creativity June*, 204.

[Smith 2011] Smith, T. 2011. One and Three Ideas: Conceptualism Before, During, and After Conceptual Art. http://www.e-flux.com/journal/29/68078/one-and-three-ideas- conceptualism-before-during-and-after-conceptual-art link. Accessed: 2018-02-17.

[Takagi 2001] Takagi, H. 2001. Interactive evolutionary computation: Fusion of the capabilities of ec optimization and human evaluation. *Proceedings of the IEEE* 89(9):1275–1296.

[Toivanen, Toivonen, and Valitutti 2013] Toivanen, J. M.; Toivonen, H.; and Valitutti, A. 2013. Automatical composition of lyrical songs. In *The Fourth International Conference on Computational Creativity*.

Harnessing NLG to Create Finnish Poetry Automatically

Mika Hämäläinen
Department of Digital Humanities
University of Helsinki
Fabianinkatu 33, 00014 Helsinki Finland
mika.hamalainen@helsinki.fi

Abstract

This paper presents a new, NLG based approach to poetry generation in Finnish for use as a part of a bigger Poem Machine system the objective of which is to provide a platform for human computer co-creativity. The approach divides generation into a linguistically solid system for producing grammatical Finnish and higher level systems for producing a poem structure and choosing the lexical items used in the poems. An automatically extracted open-access semantic repository tailored for poem generation is developed for the system. Finally, the resulting poems are evaluated and compared with the state of the art in Finnish poem generation.

Introduction

Creating poems automatically is a difficult task to tackle, especially since poetry as a genre is fragmented and not easily defined (Juntunen 2012). What makes computer generated poetry more difficult than a traditional NLG task is that poems usually express their meaning in an indirect language by means of different rhetorical devices, such as, metaphors, ellipsis, rhymes and so on. These are issues to resolve above and beyond the mere production of grammatical output that follows the syntax of the target language.

Grammaticality is a big issue especially in the case of morpho-syntactically rich languages such as Finnish. Finnish has a set of agreement and government rules in its syntax, which means that words in a sentence affect each other's morphological form depending on their syntactic relations. In other words, where as in English it is almost possible to produce grammatical output just by using words in their dictionary forms (i.e. lemmas) in a sentence with slight to no modifications at all, in Finnish, more often than not, words have to be inflected to fit the morphological requirements of the syntax in the sentence.

One of the tasks we have to solve in order to produce grammatical poems with an NLG pipeline is to build a linguistically robust system for resolving the morpho-syntax of the words in a sentence. Due to the lack of freely available systems of this kind for Finnish, we have to build one of our own. On top of this syntax producing system, we then build higher level functionality to produce poetry.

In addition to solving the challenging problem of grammaticality, in this paper, we also present a way of building a semantically linked database to use in computationally creative systems. This database consists of syntactic relations between words, which also reveals a great deal about their semantics and the intercompatibility of their meaning in a sentence. Finally, on top of the syntax generator and this semantic database, we propose a method for generating modern poetry in Finnish.

The key notion behind the system is that if the output is grammatical, i.e. the language is good, the structure resembles that of a poem and there is semantic cohesion within the poem, the poems produced by the system will be accepted as poetry by people. In this paper, we discuss how we solved these different requirements for the poetry produced.

The poem generator described in this paper replaces the previous poem generator in a computationally creative system called the "Poem Machine[1]". The generator then serves as a component in a larger whole of a system with a focus on providing an environment for human-computer co-creativity. The Poem Machine is intended to be used in elementary schools to aid school kids in writing poems of their own by removing the problem of "a blank page" and by offering a computer-generated poem as a starting point for poem writing. This paper, however, focuses solely on the poem generator as an independent component of the system leaving the co-creativity aspect of the larger system outside of the scope of this paper.

Related work

Poem generation has received its fair share of interest in past research. The problem has been approached from different angles such as translation with weighted finite-state transducers (Greene, Bodrumlu, and Knight 2010), text transformation via word embeddings (Bay, Bodily, and Ventura 2017), templates (Colton, Goodwin, and Veale 2012) and case-based reasoning (Gervás 2001) among others. In this section, we describe the approaches used in the context of Finnish poetry generation in more detail.

The current state of the art in Finnish poetry generation is the P. O. Eticus system (Toivanen et al. 2012). Unlike in our approach, P. O. Eticus does not employ a linguistically solid NLG surface generator to produce syntactic Finnish, instead it takes its syntax from existing poems. The way the system

[1] http://runokone.cs.helsinki.fi/

works is that it takes a ready-made human written poem at random, analyzes it morphologically and swaps some words in the poem with new ones making sure that the morphology, such as case and number, matches that of the words in the original poem that are to be replaced.

From a linguistic perspective, this means that the grammaticality of the final poem relies on pure chance. By making sure that the morphology matches, the system is able to solve agreement of nouns and adjectives, but will fail if the government rules for the new words are different from the ones that existed in the poem. This also means that the subject of the sentence cannot be changed at will into a different person or number without producing ungrammatical output, let alone the fact that the system is incapable of producing any syntactic structure of its own, as it relies heavily on the structure of the ready-made poems.

Another take on the poem generation in Finnish is that of (Kantosalo, Toivanen, and Toivonen 2015). This is by no means a sophisticated approach as it takes poem fragments from thematically divided children's literature at random and puts them together without any further analysis of their semantics or meter. However, it is important to showcase this approach as it was the original one in use in the Poem Machine, which nowadays uses a generator based on the one we are describing in this paper. The poems produced in this way, were more grammatical than the ones of P. O. Eticus because the verses were never altered, but they had less semantic cohesion because the verses came from very different sources, although all of them shared a keyword.

The NLG Pipeline

In this paper, we propose an NLG pipeline with independent modules to tackle different problems of generating Finnish poetry. Syntax, morphology, semantics and poetic structure are all different parts of the system. This makes it possible to separate the two goals for the generated poems at the level of implementation, grammaticality and semantic cohesion.

An NLG pipeline has traditionally been divided into four different steps: content determination, sentence planning, surface generation, and morphology and formatting (Reiter 1994). This is also the definition for NLG we are following in this paper.

In the content determination step, an input such as a query is fed to the system based on which the output will be produced. This might be for example weather for a particular city requested by the user. The results of the query are in a form of a semantic representation of the information that will be conveyed to the user in the final output. In addition to what the information will be in the final output, this step also tackles the question of how it should be communicated such as in the form of rhetorical planning.

In sentence planning, the abstract semantic representation is further processed into an abstract syntactic representation. This means that the lexical items and their syntactic relations are chosen by the system. This step works on a high level of abstraction of syntax and it does not require knowledge of how the syntax is actually produced. This means that it has no knowledge of government or agreement rules in the language, which, as we will see later in this paper, are extremely important features in the Finnish grammar.

The surface generator is the one in charge of resolving the actual syntax needed to express the abstract syntax relations (such as subject, predicate, object) by following the grammar rules of the language in question. For example, in Finnish the predicate has to agree with the number and person of the subject, and the object has to be in a case governed by the predicate verb.

Finally, the syntactic knowledge is passed to a morphological generator which is the one in charge of inflecting the lexemes chosen in the previous steps based on the lemma and the morphological features resolved by the surface generation step.

In this paper, we will focus on the two higher level steps of the NLG pipeline in *the Poem Generator* section. For the surface generation part, we have developed a tool called Syntax Maker[2] (Hämäläinen 2018) due to the absence of openly available NLG tools for Finnish. Syntax Maker is only briefly discussed in this paper, for a more detailed description and evaluation has already been published elsewhere (Hämäläinen and Rueter 2018).

The reason why Finnish requires a sophisticated tool for generating the syntax of the sentence lies in the highly agglutinating nature of the language. In Finnish, syntactic roles of words are not expressed by strict word order as is the case in English, but rather by inflecting words accordingly to their syntactic function. There are two syntactic rules that affect the word forms in Finnish: agreement and government.

Agreement in the Finnish context means that the predicate verb has to agree in number and person with the subject and that adjective attributes have to agree in case and number with the noun they modify. This has been solved by rules in the implementation of Syntax Maker.

Government is a more difficult phenomenon. Usually, in Finnish, verbs take their direct object either in the genitive (or the accusative for the personal pronouns) or the partitive. This cannot be deduced by easy rules, but rather has to be known for each verb individually. This was achieved by learning the cases of the objects (direct and indirect) automatically from the Finnish Internet Parsebank (Kanerva et al. 2014) and its syntactic bi-gram data.

The morphological generation is done by using Omorfi (Pirinen et al. 2017), which is a finite-state based tool for analyzing and generating the morphology of Finnish words. The morphological forms together with lemmas resolved by Syntax Maker are given to Omorfi, which inflects the words accordingly to its rules and the input.

The Semantic Repository

Whereas P. O. Eticus uses a graph of semantically similar words obtained by connecting words together with a log-likelihood ratio test (LLR), we want not only to capture the overall semantic relatedness of the words but also word relations in their syntactic position. In other words, we do not

[2]Syntax Maker is released as an open source Python library on https://github.com/mikahama/syntaxmaker

simply want to build our network so that we can deduce that *dog* and *dog food* are related based on their shared context, but rather that they are related by virtue that *dog* is capable of performing the action *eat* and *dog food* can serve as a direct object for such an action.

In order to capture both the semantics and syntax, we build our semantic repository[3] so that it contains lists of lemmatized words by their parts-of-speech. These lists are interconnected to a network based on the syntactic relations these words have had in a corpus. The strength of the connection is determined by the frequency of co-occurrence of the words in a given syntactic relation revealing more about the semantic relatedness of the words. To achieve this, a syntactically parsed corpus of Finnish is needed.

As a corpus for extracting the semantic knowledge, we use the Finnish Internet Parsebank (Kanerva et al. 2014), and more specifically its data-set of syntactic bi-grams. The corpus is one of the largest syntactically parsed ones in Finnish consisting of 116 million sentences and 1.5 billion individual tokens. The text of the corpus originates from different sources found on the internet by Common Crawler[4].

The data has been automatically parsed into syntactic dependency trees, and the syntactic bi-gram data consists of bi-grams of words that have appeared next to each other in the syntactic tree. This means that as opposed to a traditional bi-gram, it is perfectly possible that words that have not been immediate neighbors in the sentence, but are related to each other by one arc in the syntactic tree, appear in the bi-gram list. In other words, for example, a noun acting as a direct object of a verb will appear in the syntactic bi-grams even if in the actual sentence there was an adjective in between the verb and the noun.

We build our semantic repository based on the co-occurrences of the words in the syntactic bi-grams. Since the data-set has been parsed entirely automatically, however, the data is not free of noise. This is why, we define additional requirements for the two words of a bi-gram before we update the relation to our semantic repository.

For verbs, we use the syntactic knowledge in Syntax Maker to perform an additional check. For each verb found in the bi-grams, we query Syntax Maker for the valency, in other words, how many objects the verb can take, and the cases of the objects. We only update the verb-object relations to the repository if the noun has been inflected in a case that is possible for the verb in question, given its case government.

For subjects, it is not necessary to query the Syntax Maker because in Finnish, the subject is, almost without an exception, always in the nominative case, which means that we can check that directly. Additionally the verb has to be in active voice, because the direct genitive object of a verb in passive voice appears in the nominative, which, if not filtered out, would introduce more noise in the data.

A noun-adjective relation is only considered if the noun and the adjective share the same grammatical case. This is due to the Finnish agreement rule which requires the case of an adjective attribute to agree with the case of the noun.

For other syntactic relations such as adverb to verb relation, we don't specify any further constraints based on the compatibility of the words according to their morpho-syntax. This couldn't even be achieved, for example, in the case of adverbs and verbs, as there are no agreement or government rules for them in the Finnish grammar.

By imposing restrictions of morpho-syntax, we were able to solve a part of the issue caused by the noise in the data, that is, the true syntactic relatedness of the words without erroneous relations introduced by the parser. The syntactic parser, however, was not the only source of errors in the data. The data also contains a multitude of words that are incorrectly tagged, for example the adjective in its partitive form *esiintulevaa* (appearing) was incorrectly tagged as a verb. Also, the corpus contains non-words such as *her?nnyt* (awoken) with an encoding error, and non-lemmatized word forms as lemmas such as the optative *heitetääs* (let's throw) where the correct lemma would be the infinitive *heittää* (throw).

In order to remove incorrectly tagged words from our syntactic repository, we go through all the words with Omorfi. Omorfi produces all the possible morphological interpretations for its input word form. In the case of Finnish, this usually results in a long list of possibilities because the inflected forms of Finnish are frequently homonymous. For example, the word form *voi*, is interpreted by Omorfi as a possible form of *voi* (butter), *voida* (can), *voitaa* (to butter), *voittaa* (to win) or *vuo* (flow).

There are two things we look at in the list of possible lemmas produced by Omorfi for each word. First, we look to see if at least one of the possible interpretations has the same part-of-speech reading as recorded in our semantic repository and if the lemma of at least one of the interpretations with the same part-of-speech is the same as the word in the repository. If the part-of-speech does not match, or the word is not in a lemmatized form, we remove it from the repository. Because Omorfi is a rule based system and we are not using a guesser version of it, it makes no attempt to analyze anything it does not know. This also allows us to filter out the non-words resulted mostly from encoding errors or spelling errors.

After the extraction and filtering process, our semantic repository consists of over 9569 adverbs, 18300 verbs, 5900 adjectives and 965000 nouns that are connected to each other by the syntactic relation they have shared in the corpus weighted by the frequency they have appeared together in that syntactic relation. The high number of nouns is due to the Finnish orthographic rule of writing compound nouns together as one word, where as in English they would be written separately, for example the English *gas station* and its Finnish translation *huoltoasema*, which consists of two words *huolto* and *asema*.

The Poem Generator

In this section, we describe how the higher levels of the NLG pipeline, namely content determination and sentence planning, are implemented in the system. These operations rely

[3]The semantic repository can be browsed and downloaded on https://mikakalevi.com/semfi/

[4]http://commoncrawl.org/

mainly on the semantic repository as their main source of data, but the Finnish WordNet (Lindén and Carlson 2010) is also used as an additional data-set.

The actual poem generation part is divided into multiple diverse verse generators. Each one of these verse generators is only in charge of producing one verse of the poem. Semantic cohesion is achieved by the fact that each verse generator takes a noun as its input and outputs a noun together with the produced verse. This noun is then fed to the next verse generator. Some of the verse generators do not modify the noun while others change it. This way the verse generators produce a semantically coherent whole.

Each verse generator implements its own content determination and sentence planning steps. Regardless of the verse generator, the content determination starts with the input noun passed to it either by the user or the previous verse generator. This noun is used to look up related words in the semantic repository and additionally in the Finnish WordNet. Each verse generator has an abstract definition of the syntax of the verse in the form of syntactic relations that can be expressed by the verse. This is the sentence planning part of the generators. The actual realization of the grammar is done by feeding this information to Syntax Maker.

The decision which verse generators will be used and in which order is defined by the poem structure database. The database consists of a set of hand-coded poem structures. These structures only state the generators and their order, but they do not affect the functionality of each individual generator in any way. In theory, the verse generators and their order could also be randomized or decided automatically in a more justifiable way than pure random, but for now, we have opted for an approach involving predefined structures to ensure a higher structural coherence within the poem.

Verse Generators

In this section, we will explain the functionality of each individual verse generator in the system. There are altogether 12 different verse generators implemented in the system.

Metaphor Generator The content determination for the metaphor starts by looking at the list of verbs the input noun can act as a subject for. We want to construct a metaphor of a form *X is Y* where the two nouns, *X* and *Y*, are connected together by a verb that is semantically strongly related to both of them. The metaphor generator will also output a second verse right after the metaphorical one explaining the metaphor by revealing the verb.

By trying out different values, we reached to two threshold values for the frequency of the subject relation between the noun and the verb in the semantic repository. A verb is considered for metaphor production if it has occurred at least 20 times and at most 1000 times with the noun. This way, we can filter out verbs that aren't descriptive for the noun because they are too frequent in general and also verbs that don't co-occur often enough.

The next step is to list other nouns that can act as subjects for these verbs. These other nouns are also checked for the frequency of their subject relation to the verb. This yields us lists of possible nouns we can use in the metaphor together with the linking verbs. This, however is not enough, because now these lists will contain a lot of nouns that are semantically too similar. For example, *man is a woman* would be a frequently appearing metaphor because both of the nouns have a lot of verbs in common. This is why we remove all the nouns that have more than 5 verbs in common with the input noun we are searching metaphors for. This results in a list of nouns that are far enough semantically from each other.

Out of the obtained noun-verb list, we pick a noun and a verb at random to form the metaphor. For the second, metaphor explaining, verse we look at the number of objects the selected verb can have and fill the object slots with weighted random based on the object relations and their frequencies with the verb in the semantic repository. If the verb doesn't take an object, an adverb related to the verb is picked with the same weighted random approach as in the case of objects.

Rakkaus on luovuus	Love is creativity
Se kukoistaa ajan	It blossoms for a while
Viha on tapa	Hatred is a habit
Se ruokkii ajattelua	It provokes thought

The examples given above are possible outputs from the metaphor generator. The structure of the sentence planner is predefined as it is passed to the surface generator. The metaphor generator passes the newly picked noun to the following verse generator.

Synonym in Essive This verse generator creates sentences of a type *As a synonym, it does something*. How this is done, is that a synonym is looked up for the input noun in the Finnish WordNet by using NLTK (Bird, Klein, and Loper 2009). This is done by querying all the possible synsets for the noun and getting the lemmas for them. Then, a noun is picked at random from this synonym list to appear in the essive case in the sentence. The essive corresponds to *as a noun* in English.

The verb appearing in the verse is again looked up in the semantic repository. The verb is picked based on the noun obtained from the WordNet so that the noun can function as its subject at weighted random. The complements of the verb are filled based on the valency of the verb and the objects linked to it in the semantic repository. This produces verses of the following kind.

Passipoliisina se uskoo laatuun	As a patrol officer, he believes in quality
Konnana se rötöstelee	As a crook, he nicks

The first example was generated for the input noun *police* and the second one for *dog*. It's important to note that the synonyms coming from the Finnish WordNet might not always be truly synonyms such as *dog* and *crook* because the Finnish WordNet has been translated from the English one directly and sometimes the Finnish translations are quite far from the English originals. This generator passes the WordNet synonym to the next generator.

Rhetorical Question This generator looks up an adjective for the input noun in the semantic repository based on the noun-adjective attribute relationship. The adjective is picked at weighted random based on the frequency of co-occurrenceIn addition, an interrogative pronoun is picked at random to form the question. The idea behind this question forming generator is that if we know that the noun has the adjectival property we are forming the question of, we can presuppose this. Instead of stating something is something, when we know that it'the case, we can just ask *why* that is the case, *how* and so on.

Milloin liekki on keltainen?	When is a flame yellow?
Miten paha on uhka?	How bad is the threat?

The examples illustrate verses generated for *flame* and *threat* respectively. The input noun is passed as such to the following verse generator.

Personal Pronoun Verses This category consists of four different generators. What they have in common is that they use a 1st or 2nd person personal pronoun either in the plural or singular as their subjects. The simplest one of them just looks up an adjective based on the input noun and forms a question of a type: *am I adjective?*.

There are two verse generators that are used to express an attitude. The first one forms a main clause with a personal pronoun subject and a verb that can be used to express an attitude, such as *hope* or *doubt*. The main clause can be turned into negative with a 50 % chance or additionally into a question with a 50 % chance. After producing the main clause, a subordinate clause is added to the main clause. The subordinate clause takes the input noun as its subject and then proceeds into looking for a suitable verb and objects and an adverb for it in the same manner as described for the previous verse generators.

The other attitude expressing verse generator picks a verb based on the pronoun picked to generate the verse. This is also done by a weighted random based on the subject connection of the verbs and the pronoun. Then it uses the input noun as an object for this verb. In order to express an attitude, an auxiliary verb is selected at random to be used in the sentence.

The last personal pronoun verse generator formulates a conditional subordinate clause in which the input noun is the subject and the verb and their objects are picked as seen before. The main clause has a verb in the conditional mood with a personal pronoun as its subject, the verb picked is selected on the basis of the input noun. This is done in such a way that the verb takes a noun that can act as its object according to the semantic repository.

Olenko huima?	Am I wild?
Emmekö me ajattele, ennen kuin nokkeluus pelastaa maan?	Won't we think before cleverness saves the earth?
Minä haluan noudattaa silkkiä	I want to follow silk
Vaikka savut houkuttaisivat onnenonkijaa, käyttäisitkö sinä savua?	Even if the smokes lured a fortune hunter, would you use smoke?

The examples above are output from the verse generators in the order of their presentation in this section. These are produced for the nouns *week*, *cleverness*, *silk* and *smoke* respectively. None of these verse generators alter the input noun, but rather pass it as it is to the following generator.

Paraphrase The paraphrasing verse generator expresses the input noun in other words by looking up a suitable hyponym or hypernym for it in the Finnish WordNet. In addition, an adjective is picked for the input noun from the semantic repository at weighted random. This adjective is used to describe the noun obtained from the Finnish WordNet in the verse.

Vesi, tuo ihmeellinen neste	Water, that wondrous liquid
Rukous, tuo hiljainen siunaus	Prayer, that silent blessing

The examples above are generated for *water* and *prayer*. The input noun is added to the beginning of the verse, separated by a comma. The verse generator passes the noun looked up in the WordNet to the next verse generator.

Relative clause This verse generator creates a relative clause which takes the subject position of the main clause in the verse. The object of the relative clause is a synonym for the input noun based on the Finnish WordNet. The object of the main clause is the input noun. The verbs for both clauses are looked up from the semantic repository based on the nouns they will have as objects.

Se, joka loppuu hauskuuteen, kertoo ilosta	What ends in fun, tells of joy
Se, joka lukee surun- valittelua, kuuluu suruun	Who reads condolence, forms part of the sorrow

The examples above are generated by using *joy* and *sorrow* as their input. The verse generator passes the input noun to the following generator unmodified.

Simple generators There are two extremely simple verse generators in the system. One is used to address a noun. What it does is that it outputs the input noun followed by a comma. The other simple verse generator generates tautologies, either in indicative or in potential, of a type *Xs are Xs*.

Poika,	Boy,
Pojat ovat poikia	Boys will be boys

The above examples are output for *boy* in both generators. Neither of them swaps the input noun, but rather passes it on as it is.

Example Poem

Here is an example poem to illustrate how the different verse generators can play together in a poem structure.

Usko on ihminen	Faith is a human
Se pelastuu kuolemasta	It is rescued from death
Henkilönä se upottaa veteen	As a person, it sinks in water
Miten arvoinen on henkilö?	How valuable is a person?
Minä en kartuta henkilöä	I don't accumulate a person
Olenko hieno?	Am I elegant?

Evaluation

In this section, we conduct an evaluation of the poems produced by the system. Evaluating poetry, even in the case of a human produced one, isn't an easy task, and it is something that is very difficult to do objectively. This is why we didn't want to come up with an evaluation metric of our own, rather we chose to use the same subjective evaluation metric that was used for P. O. Eticus (Toivanen et al. 2012). An additional advantage of this is that we can see how well our system fares in the same evaluation as the state of the art.

The P. O. Eticus was evaluated by 20 randomly picked university students. In order to have a better comparability of the results, we also randomly recruited university students for our evaluation. In the evaluation, we randomly selected 5 poems produced by our generator and 5 poems by the poem fragment approach (Kantosalo, Toivanen, and Toivonen 2015) which was previously in use in the Poem Machine. We had altogether 25 evaluators to go through poems produced by both systems. The order in which the poems were presented to the evaluators was randomized.

The evaluators were asked to evaluate texts rather than telling them directly that the texts are supposed to be poems. They weren't told that the texts they were reading were computer produced.

The evaluators were asked to answer to a binary question with a yes/no answer whether the text they were reading was a poem. In addition to that they were presented with six ad ditional questions: (1) How typical is the text as a poem? (2) How understandable is it? (3) How good is the language? (4) Does the text evoke mental images? (5) Does the text evoke emotions? (6) How much does the subject like the text? These questions were evaluated in the Likert scale from one (very poor) to five (very good).

As the evaluation questions are highly subjective and the evaluators' opinions on the poems vary a great deal, the results obtained for our approach and the poem fragment approach aren't directly comparable with those obtained previously for P. O. Eticus. However, the results P. O. Eticus got when it was evaluated are shown in the chart for reference purposes.

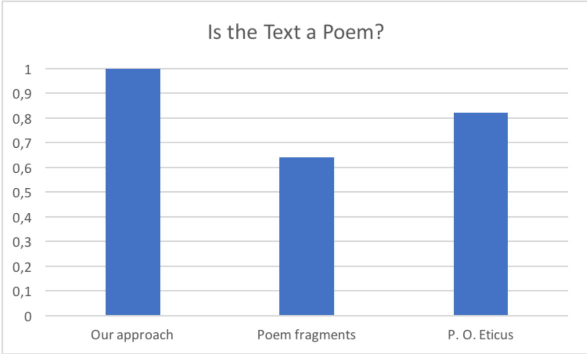


Figure 1: Results for the binary question

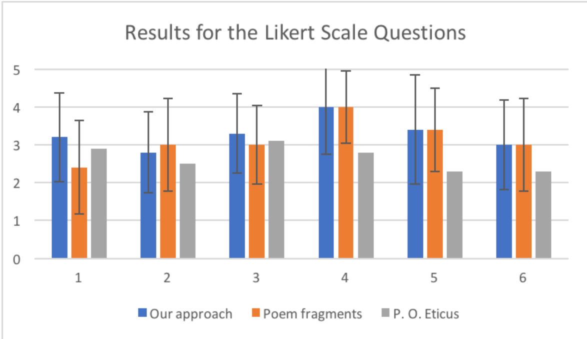


Figure 2: Results for the Likert scale questions with standard deviation: (1) How typical is the text as a poem? (2) How understandable is it? (3) How good is the language? (4) Does the text evoke mental images? (5) Does the text evoke emotions? (6) How much does the subject like the text?

Figure 1 represents the results obtained in the the first binary question whether the output of the generators was considered as a poem. An important finding is that the poem fragment approach only generates output recognizable as poetry 64 % of the time whereas the judges agreed unanimously that the output produced by our method is poetry. This is probably due to the fact that the poem fragment approach doesn't aim towards a poem-like structure whereas our approach uses predefined poem structures.

The results for the Likert scaled questions are shown in Figure 2. The results show that our approach outperformed the existing poem fragment generator in the typicality aspect and in how good the language was. Based on these results, we can deduce that our system is capable of producing poetry that is also accepted as poetry. Also the grammatical correctness of the output is high enough to score well against the fragment approach which essentially uses human written fragments.

Understandability is the only parameter in which the poem fragment approach performed better. This is could be due to the fact that the fragments are written by humans, which might contribute to them being easier to understand, where as the words picked by our system to be used in a verse, might result in a sentence that is semantically more difficult to grasp.

The fact that our system seems to fare well in comparison with the state of the art on all aspects seems promising. However, since the poems generated by both systems were evaluated by different people, further study is needed to draw any final conclusions on which one actually performs better in this evaluation setting.

Discussion and Future Work

The generator discussed in this paper is a first step towards an NLG pipeline in poem generation in Finnish. Now that the most difficult parts of producing Finnish have been solved, namely the rich morphosyntax of the language, and that we are capable of producing grammatical Finnish from abstract syntactic representation, the next step is to reduce the hand-crafted nature of the verse generators. We are currently looking into the possibility of learning verse structures from real poetry into an abstract syntactic representation that we could fill, for instance, with the content determinators already defined for the individual verse generators. This would mean that we would only need to replace the sentence planning part of our pipeline to introduce more freedom into the system.

We could also extend the semantic repository not only to contain a wider list of syntactic relations but also to contain semantic data of a different nature. This could, for example, mean linking related words based on word embeddings. The extension of the semantic repository is a requirement we have already found in our initial experiments of using learned verse structures, because the syntactic relations in real verses are more complex than the ones modelled in our semantic repository.

Another important aspect for future research is studying this method in the context of the system for which it was initially built, namely the Poem Machine. Studies are currently underway on the co-creativity aspect of the Poem Machine in which the method described in this paper is in a collaborative setting with school kids assisting them in writing poetry of their own.

Conclusion

In this paper we have presented and evaluated an NLG approach for poem generation for the morphologically rich Finnish. The proposed approach is currently in use in a computationally creative system called Poem Machine which makes human computer co-creativity possible. The results of the evaluation seemed promising and we identified future directions for this research.

As a result of the study, an open source surface generation NLG tool for Finnish (Syntax Maker) was publicly released. Also, the syntactic repository data set has been made openly available for anyone interested in building their work on top of it.

Acknowledgments

This work has been supported by the Academy of Finland under grant 276897 (CLiC).

I thank my PhD supervisor Jack Rueter for a keen eye on English grammar while reading this paper and his enthusiasm in bringing this approach to the minority language context.

References

Bay, B.; Bodily, P.; and Ventura, D. 2017. Text transformation via constraints and word embedding. In *Proceedings of the Eighth International Conference on Computational Creativity*, 49–56.

Bird, S.; Klein, E.; and Loper, E. 2009. *Natural Language Processing with Python*. O'Reilly Media.

Colton, S.; Goodwin, J.; and Veale, T. 2012. Full-face poetry generation. In *Proceedings of the Third International Conference on Computational Creativity*, 95–102.

Gervás, P. 2001. An expert system for the composition of formal spanish poetry. *Knowledge-Based Systems* 14(3):181–188.

Greene, E.; Bodrumlu, T.; and Knight, K. 2010. Automatic analysis of rhythmic poetry with applications to generation and translation. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, 524–533. Stroudsburg, PA, USA: Association for Computational Linguistics.

Hämäläinen, M., and Rueter, J. 2018. Development of an Open Source Natural Language Generation Tool for Finnish. In *Proceedings of the Fourth International Workshop on Computatinal Linguistics of Uralic Languages*, 51–58.

Hämäläinen, M. 2018. Syntax Maker. https://doi.org/10.5281/zenodo.1143056.

Juntunen, T. 2012. Kirjallisuudentutkimus. In *Genreanalyysi: tekstilajitutkimuksen käsikirja*, 528–536.

Kanerva, J.; Luotolahti, J.; Laippala, V.; and Ginter, F. 2014. Syntactic N-gram Collection from a Large-Scale Corpus of Internet Finnish. In *Proceedings of the Sixth International Conference Baltic HLT*.

Kantosalo, A.; Toivanen, J.; 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.

Lindén, K., and Carlson, L. 2010. FinnWordNet-WordNet på finska via översättning. *LexicoNordica Nordic Journal of Lexicography* 17:119–140.

Pirinen, T. A.; Listenmaa, I.; Johnson, R.; Tyers, F. M.; and Kuokkala, J. 2017. Open morphology of Finnish. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics, Charles University.

Reiter, E. 1994. Has a Consensus NL Generation Architecture Appeared, and is it Psycholinguistically Plausible? In *Proceedings of the Seventh International Workshop on Natural Language Generation*, INLG '94.

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*.

An Evaluation of the Impact of Constraints on the Perceived Creativity of Narrative Generating Software

Lewis Mckeown
School of Computing
University of Kent
Canterbury
lam54@kentforlife.net

Anna Jordanous
School of Computing
University of Kent
Medway
a.k.jordanous@kent.ac.uk

Abstract

This work investigates the impact of constraints on the perceived creativity of the output of narrative generating systems, in order to understand what level of constraint application results in the most creative output. To achieve this, software is written that generates short stories, using adjustable levels of constraint meant to reflect those utilised by other narrative generating systems. These systems are presented at different positions along a spectrum, which we posit arises from the application of constraint. The creativity of the output is then assessed by human evaluators. The results are promising and show a clear variation of response based on the level of constraint imposed on the narrative generation process. The results show a sweet spot for maximal creativity closer to the less constrained end of the spectrum, which demonstrates the potential for more creative software by the relaxing of constraints.

Introduction

Traditionally narrative generation systems have focused on producing outputs that can be proven to adhere to a particular narrative theory (Pickering and Jordanous 2016) or that greatly restrict the scope of what is possible (León and Gervás 2008) to ensure what is sometimes referred to as appropriateness (Sharples 1996). This project removes some of the first order assumptions about what constitutes a coherent narrative in an attempt to see if these constraints are in fact a hinderance to the creative potential of software.

It is evident that constraints have an effect on creativity, though it is less well established what that effect is. Computational creativity has identified the importance of being able to operate with the freedom to identify and transform constraints. Boden's three-fold model of creativity includes transformational creativity: the explicit "transforming" of constraints that determine the space of possible creative outputs (Boden 2004). One of the key components of creativity highlighted by Jordanous & Keller (2016) is "Independence and Freedom", defined in part as the ability to "challeng[e] cultural or domain norms". Discovering if challenging these norms is best achieved by the removal or imposition of constraints is one of the goals of this project.

Reviewing existing narrative generating software highlights a common focus on putting together story components in ways that comply with a predefined set of constraints (Pérez y Pérez 2015; Laclaustra et al. 2014). This can generate comprehensive and sensible stories that demonstrably adhere to a particular narrative structure. However it can lead to generic narratives which may fail to incorporate the potential for the bizarre or the unexpected often associated with high levels of creativity. Yet systems that remove these constraints may appear too unmoored from existing cultural norms to be considered creative. It is this dichotomy that motivated the study of constraints in particular, namely; how can software deviate from these norms whilst still demonstrating awareness of them?

By adjusting constraints that are often taken for granted when developing a narrative generation system, their impact can be assessed and the extent to which they may or may not affect the attainment of transformational creativity as defined by Boden (2004) can be measured. Our results show that a more relaxed narrative generation system can produce works that are perceived to be more creative. This is a fine line, as too relaxed or too moderate applications of constraint will result in much lower creativity ratings. If a system strays too close to randomness the perceived creativity will be heavily penalised. In contrast the strictest application of constraints demonstrated the second highest level of creativity when evaluated by humans.

In the following sections a selection of existing systems are presented and shown to represent a spectrum of constraint application. How this spectrum motivates the choice of evaluative methods is then discussed, followed by an introduction to the software that was written to test the impact of constraints and the data gathering methods, finally a detailed discussion of the findings is presented.

The Spectrum of Constraint Application in Narrative Generating Software

The software presented here spans a range, starting with context aware and goal oriented systems which can vary significantly in their application of constraint, to writing systems at what might be considered the other end of this spectrum; in which no character or context awareness could reasonably said to be present, but a large corpus of existing texts forms the knowledge base from which the software learns and generates new artefacts.

Carlos León and Pablo Gervás made the storytelling system CAST to generate narratives based on the exploration and transformation of constraint rules (León and Gervás 2008). CAST starts with a knowledge base of facts and a set of constraints on how those facts can be combined. It then works to combine the facts in a way that is considered coherent given the constraints in place. This might involve considering a sequence of actions like

$$kill(criminal, policeman) \rightarrow eat(policeman, sandwich) \quad (1)$$

as invalid, as the dead policeman can not eat a sandwich.

Simply combining facts however will not lead to satisfying or creative output, it could at best achieve Boden's combinational creativity in a naive sense. The authors acknowledge this and attempt to circumvent it by ensuring the knowledge base evolves with each combination of ideas. They even go so far as to say that allowing a small number of non valid states to be used can lead to an increase in creativity. A point that is not touched on much, but hints at Sharples' insistence that breaking constraints will likely enhance creativity (Sharples 1996). However Sharples also stated that the application of constraint is necessary, to ensure that what is generated does not become "a ramble of nonsense". This fact appears to have influenced the authors more as they are keen to avoid the generation of narratives that might be considered "partial" or "non coherent" (León and Gervás 2008), perhaps imposing a constraint on the system that might limit the potential for radical originality.

Pérez y Pérez's system MEXICA (Pérez y Pérez 2015) generates stories about the inhabitants of ancient Mexico City using the engagement, reflection model of narrative generation. This model involves a process of generation called engagement, in which MEXICA combines contexts from its knowledge base, looking for similar contexts to put together. This is followed by a process of reflecting and criticising the work developed so far, checking that preconditions can be satisfied and attempting to evaluate novelty. The goal is to avoid creating narratives that are too similar to existing stories in the agent's knowledge base or stories which do not adhere sufficiently to the Aristotelian narrative structure, thus ensuring novelty and creativeness (Pérez y Pérez 2015; Pérez y Pérez and Sharples 2004). However given the imposition of an established narrative structure that must be maintained, and the avoidance of certain factors such as repetition or similarity, even though a certain amount of adaptability is inherent in the engagement reflection model, it is still very constrained in the amount of transformation or exploration that will be permitted.

DAYDREAMER utilises a relaxed planning mechanism to guide the actions of a daydreaming agent. Mueller and Dyer explore the utility of daydreaming in machines, attempting to provide a computer model for daydreaming that will generate short stories. They posit that the relaxed constraints of the daydreaming mind can facilitate the exploration of possibilities that would not normally be pursued, which can in turn allow for the exploration of unusual or not often linked ideas (Mueller and Dyer 1985).

Mueller writes that

> There are certain needless limitations of most present-day artificial intelligence programs which make creativity difficult or impossible: They are unable to consider bizarre possibilities and they are unable to exploit accidents.
> (Mueller 1990 P.14)

This is a rather novel approach in the domain of narrative generating systems, which often focus on adherence to an established narrative structure or literary theory. Actively seeking the bizarre or the accidental discovery of new combinations of ideas seems far more likely to generate creative works. To achieve this there must be some level of constraint to ensure appropriateness but the extent to which other aesthetic or structural facets of narrative are required is greatly reduced by DAYDREAMER.

There are still, however, defined goals involved that the daydreaming agent works towards and there is little discussion of adjusting the constraints imposed by the relaxed planning mechanism. This is a constraint that few creative systems that produce a narrative seem willing to break.

Benjamin (Sharp and Goodwin 2016) is a long short-term memory recurrent neural network that has developed several screenplays, like Sunspring (Benjamin 2016). Unlike the other systems discussed, Benjamin works without agents trying to achieve goals, or sets of facts that ensure consistency. Using a large corpus of existing screenplays it can be trained to learn and develop its own narratives in a style learned from the texts provided (Sharp and Goodwin 2016). This is an application of deep learning that has been applied successfully before in creating artistic works with an aim of learning and maintaining a structure.[1]

Developing story telling software that is not explicitly tasked with creating characters and managing their interactions is quite far removed from other systems discussed up to this point and its results are vastly different. They certainly would not be highly rated by Pérez y Pérez's implementation of the three layers model (discussed in the next section) and would likely be considered incoherent by the standards of CAST. However without the level of constraint implicit in the requirements for characters with

[1] An LSTM RNN has been used to learn and compose its own blues licks with a particular focus on structure (Eck and Schmidhuber 2002).

predetermined goals, Benjamin's output could have the potential for far more unusual or bizarre ideas. There is the ability to exploit accidents, though perhaps not in the way intended by DAYDREAMER's authors. The model and the steps used to arrive at the output may be more opaque than the other systems, but the results and methods could be considered closer to a truly generative act than other more structured or constrained systems. The curation coefficient of the programmers is less obvious and the results will likely provide more of the *shock* of surprise Boden anticipates when seeing something truly creative (Boden 1998), as even knowing the corpus provided, the resultant artefacts are unlikely to be something the programmers would have predicted.

The variety of responses from artistic works made by neural networks definitely shows the potential for an AI system developing an aesthetic modality that is distinct from that of humans, and it is arguably transforming the conceptual space with its abstract approach to generating text. However, Benjamin's works are the most likely of the systems discussed so far to be accused of becoming a "ramble of nonsense" by Sharples. This could perhaps be countered with a discussion of the audience and creative aspects systems like Benjamin are trying to attract and replicate. However given the variation from the corpus, it is clear that there may be more than a different aesthetic taste separating Sunspring from A New Hope.

There is undoubtedly time for art like this to establish itself and maybe even provide Boden's vindication of AI creativity, but it seems that right now some constraint in the form of context awareness may help improve the public opinion of this esoteric approach to narrative generation. This motivated the search for an application of constraint which would illicit the highest rating of creativity from audiences, when compared to other positions on the spectrum.

Evaluating Computational Creativity in Narrative Generating Software

The following section covers two methods for evaluating the creativity of narrative generation systems and compares their potential utility for assessing the impact of constraints on creativity as well as assessing creativity in general. The aim is to highlight how some methods of evaluation may be biased in favour of systems towards the more constrained end of the spectrum. As many evaluation techniques focus on an adherence to a predefined structure or use other criteria that would be unfavourable methods of assessment given the foundational assumptions this paper aims to challenge.

Pérez y Pérez developed the *Three Layers* approach to evaluating computer generated narratives to give the MEXICA plot generator the ability to assess its own output and the output of other writers. The model generates a score for the plot that can be used to quantitatively assess its potential creativity (Pérez y Pérez 2014).

Layer 0 of the model involves checking for required-characteristics which are fundamental for something to be considered as having a plot. This layer does not contribute to the overall score, but a failure to meet the requirements of the model (due to unfulfilled preconditions or similarity to existing stories), will result in no evaluation taking place as the next two layers will not be completed. Layer 1 assesses the core characteristics of a narrative. Checking for the presence of climax, closure and unique or novel structures. The final layer deals with what Pérez y Pérez calls *enhancers* and *debasers* and it looks for aspects of narrative structure that, if missing, would be noticed immediately as their presence is taken for granted. Pérez y Pérez calls these *preconditions* and their absence is penalised (Pérez y Pérez 2014). Repeated sequences are also penalised and reintroducing complications is considered an enhancer. Once the narrative has been evaluated by all layers of the model a score can be provided for each layer based on the presence or absence of these valued features and the way they are structured.

This method is not without its flaws, the most glaring of which is that the idea of automating the quantitative assessment of the creative worth of an artefact is highly suspect. The model requires a level of human curation in the selection of required characteristics for layer 0 and layer 1 focuses on the inclusion of features like climax chosen by the author. Pérez y Pérez says that "a narrative without climax is not a story" (Pérez y Pérez 2014 P.5); a highly subjective statement that relies on aesthetic taste rather than some quantitative measure of worth. It almost appears that in an attempt to remove the human component from the evaluation of works, the imposition of one humans judgements has been automated. Although the layers of the model can be tweaked, the same issue will likely remain, that the criteria will be chosen by one or a small group of people and are relatively inflexible once in place. This model might be seen as imposing constraints in a way that penalises variation from expected norms, in light of this it is unlikely to value the transformation of the conceptual space in a way that might result in unusual or new aesthetic modalities.

Rather than try and skirt the need for human subjectivity in the evaluation of creativity then, it might be better to embrace it. In an earlier paper Pérez y Pérez and Sharples wrote some criteria for presenting narrative software for evaluation. They highlight that a common difficulty when assessing story generation systems is the lack of an agreed upon comparative structure (Pérez y Pérez and Sharples 2004). To solve this they proposed some rules for evaluation stating that

- The programs knowledge base should be available for human evaluation in a sensible form.
- The type or aspects of creativity being modelled should be stated clearly by the designers, as should the audience.
- The program should be capable of generating a minimum

of ten stories, 3 of which can be selected by the designers for human evaluation.

- The selected outputs should be "judged for overall quality, originality and interestingness by independent raters" (Pérez y Pérez and Sharples 2004 P.15).

This model is less programatic and perhaps harder to implement than the three layers. However it allows for a range of creative opinions to be included in the evaluation of the works by having multiple individuals assess them, rather than implementing the automated checking of criteria. The less prescriptive approach can also be considered an advantage, as it may appear prima facie to be less quantitative to have output judged by humans; quantifying these ratings is possible and their individual approaches to creative assessment could even be documented alongside their responses to provide further context to each evaluation.

When evaluating the output of creative software, the use of human participants may not be ideal for uniform data gathering, but it may represent the state of the art when assessing the novelty or creativity present in a work of art.

The Software

To support the investigation into the significance of constraints on narrative generation, some story writing software was developed. It was designed to produce narratives with a dark and dreamlike theme, taking inspiration from authors and creators like David Lynch, David Foster Wallace and Haruki Murakami among others. The goal was for the stories to replicate some of the style of magic realist or surrealist authors and auteurs, who demonstrate a high level of creativity whilst seeming to balk at the traditional constraints of narrative theory.

Early versions of the project considered using a combination self evaluation and human evaluation to assess the creative worth of generated stories. The output was to be generated by software that was using a relaxed ruleset that would be some combination of the algorithms and approaches commonly found in machine learning applications and the context aware, agent driven models such as DAYDREAMER and MEXICA. The original aim was to model the less constrained end of the spectrum, working under the assumption that a less restricted narrative generation system would be able to demonstrate creativity closer to that of the artists that inspired the project. Self evaluation was abandoned in favour of structured human evaluation and the project became more directed once the research question was narrowed, to focus on the impact of constraints in particular.

Once the decision was made to focus on constraints the evaluation strategy and implementation became key areas of focus. Initially the software's output was to be evaluated via comparison alongside other story writers which used differing levels of constraints such as CAST or DAYDREAMER. However due to the variety in style, length and availability of the other software's output it quickly became clear this was an unfit method of assessment.

To establish a more rigorous way of evaluating the output, the decision was made that it all come from the same creative agent. The software would use the same knowledge base of characters and actions to generate narratives but do so with differing levels of constraint in place. Rather than using a relaxed ruleset that was a fixed set of constraints representing some position on the spectrum of creative writing software, these constraints should be adjustable by the user. The software's output could then be evaluated by humans with a version of Pérez y Pérez and Shaples' 2004 testing methodology.

The final version of the software allowed the algorithms that dictated, character actions, locations and events to be selected prior to generation, along with other critical aspects such as whether character death would remove them from the narrative or if duplicate characters could appear. This facilitated the generation of a selection of stories which could represent positions on the hypothesised spectrum of constraint, on which human evaluation could be carried out.

Evaluation Strategy

To evaluate the output of the software a selection of Pérez y Pérez and Sharples' 2004 benchmarks for assessing story generation systems were used to develop an evaluation strategy based on user feedback.

The benchmarks recommend stating the aspects or style of creativity that the software is attempting to model, as well as the audience it is aimed at. They also recommend that the software be capable of generating at least 10 stories, and that 3 of these could be selected by the software authors for human evaluation.

This model was adhered to very closely, with the final evaluation strategy involving 10 narratives being generated and 3 selected for evaluation. This process was repeated for 4 differing levels of constraints, for a total of 12 stories which required evaluation. Before being presented with the stories, an explanation of the project, its creative aims and the target audience were provided to the respondent. These were presented alongside two dictionary definitions of creativity, focused on the production of artefacts demonstrating unusual or non traditional ideas, which served as a guide allowing users to comfortably answer to what extent they believed each story demonstrated creativity (Jordanous 2013), using the following scale

Strongly Agree: 2

Agree: 1

Neutral: 0

Disagree: -1

Strongly Disagree: -2

The respondents were also asked to indicate if they liked each story. This was primarily to separate opinions about whether creativity was being demonstrated from any other value judgements about the quality of the text.

Creating the Datasets

The options chosen to generate the datasets were developed to try and reflect a section of the spectrum of constraints used when generating narratives with software. The breakdown can be seen in Table 1.

The lowest level, dubbed *unconstrained*, was chosen mostly based on randomness, to represent the least amount of constraint a narrative could be generated with, this was meant to mimic an amount of context awareness at the level of an untrained neural network, and given the software's design, the curation coefficient likely played a large part in the resultant narratives, rather than an application of what might be deemed computational creativity by Pérez y Pérez's definition; *c-creativity* which requires the generation of new and relevant knowledge (Pérez y Pérez and Sharples 2004).

The other end of this spectrum was as *tightly constrained* as the software could be, with actions chosen by character motivation, a Markov model used to select events and locations and the options to respect character death and prevent doppelgangers being imposed.

The two middle datasets represented as *moderately constrained* were generated with a very similar set of options, the key difference being the choice in *set 1* to select events randomly rather than using a Markov model, making it slightly less constrained than *set 2*. Event choice and ordering is a non trivial aspect of any narrative and this could provide a significant impact on the resulting output. The aim was to represent more middling levels of the spectrum, with *set 1* hopefully mimicking DAYDREAMER's less constrained and more esoteric approach to event choice, whilst still imposing a constraint over action choice. As a differentiator *set 2* imposed a slightly stricter logic, perhaps more reminiscent of CAST's pursuit of coherence. Although the artistic style, approach to generation and undoubtedly, the output of each dataset was different to all of the systems used as inspiration; this should present an abstracted and high level representation of how constraints used in story writing systems can affect their output. The extent to which this is the case is discussed in the next section.

Getting Respondents

Initial evaluations were completed by a small set of people - unfamiliar with the field of computational creativity - who provided more detailed feedback and discussion following the completed assessment. Once these evaluations were completed, a post was made on the Computational Creativity Google group asking for respondents. This provided more discussion of the work and feedback gathering approach as well as a host of new respondents, 10 more complete responses in total.

Analysis of Responses

A total of 202 evaluations were received during a one week run of feedback gathering, resulting in an estimated 16 respondents. Not all respondents completed the entire survey however, so some outliers were left that needed to be removed. Despite the presence of incomplete responses a trend developed early on and remained rather consistent throughout the evaluation process. The datasets representing *tightly constrained* and *moderately constrained set 1* were consistently deemed more creative than *moderately constrained set 2* or the *unconstrained* set. This trend continued, with some minor fluctuation; *moderately constrained set 2* and *unconstrained* jumped between being deemed uncreative and simply neutral, ultimately ending up with *unconstrained* being evaluated as slightly less creative (see Figure 1).

The Impact of Constraints on Perceived Creativity

The relative unsuccessfulness of the most aleatory dataset, *unconstrained*, shows that Sharples' insistence on appropriateness and its pursuit by software representing the more constrained end of the spectrum such as CAST and MEXICA is thoroughly justified. Even given the type of creative endeavour that the project was attempting to emulate - surrealist and magic realist authors, known for bizarre juxtaposition in their work - the outputs generated using only random combinations of story components were consistently deemed less creative and liked less than their more constrained counterparts.

The *tightly constrained* dataset, the end of the spectrum in which every thing that could prohibit randomness was in place, showed the second highest level of creativity according to respondents and was liked the most. The potential for more clear character arcs, as this was the only dataset using character motivation, may help identify its popularity. One respondent in more detailed feedback even correctly identified a story from this dataset as showing evidence of character motivation. This might lead to an audience seeing more familiar tropes such as revenge or love and associating the output with works they have a clear mental model for and enjoy. The issue of conflating a positive response to the work with the presence of creativity is discussed in the next section.

The most interesting results came from the juxtaposition of the two middle datasets. With *moderately constrained set 1* rating the most creative of all four datasets, whereas *set 2* was consistently among the lowest creativity ratings, scoring just higher than *unconstrained* once the outliers were removed.[2]

The only difference between the two middle datasets was the choice of event being made randomly by *set 1* and by Markov model in *set 2*. This distinction could represent

[2]With outliers removed, this was revealed to be the fault of one story significantly affecting the (low) average. See co880.lewismckeown.com.

Datasets	Action Choice	Event Choice	Location Choice	Respect Death	Allow Doppelgangers
Unconstrained	Random	Random	Random	False	True
Moderately Constrained (Set 1)	Markov	Random	Random	True	False
Moderately Constrained (Set 2)	Markov	Markov	Random	True	False
Tightly Constrained	Character Motivation	Markov	Markov	True	False

Table 1: Breakdown of the options used to create each dataset for user evaluation.

a violation of constraint in the ideal sense that Sharples writes about, in a way that may facilitate radical originality whilst maintaining appropriateness.

Event choice is significant, however, the most constrained narrative generation systems focus primarily on the restriction of character action to ensure an arc or predictable response to stimuli. Beyond this, perhaps there is a lot of room for manoeuvre when developing what happens around characters. The potential for the bizarre with a less constrained selection of events and locations is greatly increased and may result in a potential transformation of the conceptual space, when juxtaposed with more considered character interactions. It would be charitable to attribute a level of Boden's transformational creativity to this project, but it should demonstrate the importance of a proper assessment of constraint to finding a computer model for transformational creativity.

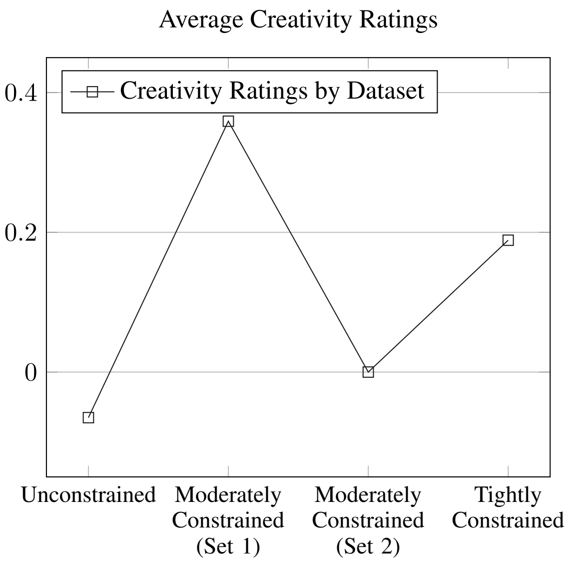


Figure 1: Average creativity ratings for each dataset.

Aesthetic Taste and Perceived Creativity

In keeping with the thoughts of some respondents, when people indicated they liked a story, this was often accompanied with a positive creativity rating. With only 8 responses indicating that they liked a story that they considered not to be showing creativity and 16 responses indicating that they disliked a story that they agreed demonstrated creativity. This is opposed to the 64 responses indicating a story was liked and demonstrated creativity and the 49 indicating a story was not liked and did not demonstrate creativity.

The choice to ask how creative each story was separate from whether a respondent liked it was primarily to remove assessments of quality or personal preference from judgements about creative merit. However given the link between creativity ratings and respondents liking the story, it seems quite likely that the aesthetic tastes of the evaluator play a large role in their assessment of an artefact's creative worth. This has interesting implications for Colton and Wiggins, who indicate that a creative machine may have different aesthetic tastes to humans. It highlights the difficulty of machines being considered creative without first mimicking existing human aesthetic standards. To provide aesthetic measures with which to assess their work or a commentary on the motivations behind it then, as Colton and Wiggins suggest (Colton and Wiggins 2012), may be a crucial step for creative machines to both achieve creative independence and be judged as having done so by human evaluators.

Summary of the Data

Overall the stories were deemed more creative than not, and respondents liked and disliked them in almost equal measure. The ratio of stories which were liked to those that were disliked could be attributed to the niche narrative style and sources of data that the software used.

The higher presence of creative to not creative output is promising for the software and any future development it might undergo. It also demonstrates the rich creative potential that non traditional and surrealist works present to creative systems. This may reflect a similitude between human made surrealist art and AI generated works.

The results (shown in Figure 1 and co880.lewismckeown.com), demonstrate that works without any effort to retain appropriateness as defined by Sharples (Sharples 1996) may result in unfavourable creativity ratings, as seen by the response to the *unconstrained* dataset. In contrast pursuing appropriateness, as the *tightly constrained* set did, demonstrably improved perceptions of creativity by human evaluators. This disproved an early hypothesis which assumed that less restriction imposed on the narrative generation process would result in higher creativity ratings for the resulting artefacts.

The most exciting finding, is that striking a balance between the pursuit of appropriateness and the breaking of constraint, may lead to far higher creativity ratings, hopefully demonstrating the significance of constraint application in any attempt to model transformational creativity as described by Boden (Boden 2004).

Evaluation

The project was ultimately an investigation, so any feedback and data returned would constitute some form of success. The interesting conclusions that the data supports and the number of encouraging responses, however, made the investigation both satisfying and rewarding. Despite this there are several areas in which improvements could be made, particularly with regards to the testing methodology and feedback gathering.

Several users commented on the repetition present in the evaluated stories. They were generated from a knowledge base consisting of only 34 actions, 25 locations and 24 events. This could have been increased to reduce potential fatigue of the users, as it could affect their ratings, particularly later in the process. The story order could also have been randomised rather than fixed, although for individuals this would make no difference, for the results as a whole it might have reduced the chance of later stories being rated as less creative because of perceived repetition. Although given the creativity ratings seen in Figure 1 it appears this did not happen, with a larger dataset it would have been prudent.

In some feedback the genre of stories was criticised for perhaps letting a less cohesive work be presented as a completed one.[3] However the project was intentionally developed in a way such that the subversions - for the most part - were intentionally done. There were toggles set when adjusting constraints before generating a story that would allow a character who had just died to have dinner with their murderer and a toggle that would allow two identical characters to go on a road trip. The intention being to knowingly subvert traditions and the expectations of readers (with an option to retain the more logical outcomes by adjusting the constraints) rather than to merely stumble into incoherence. The fact that randomly arriving at creativity is unlikely is also supported by the consistently low creativity ratings of the *unconstrained* dataset and the higher ratings of the more tightly developed datasets. So, although this is an understandable criticism, it is hopefully addressed sufficiently here and in the preceding sections.

Another similar criticism received from users, was to what extent the output could be considered a story, or if it was insufficiently fleshed out to be one. The stories were presented as short vignettes before asking for evaluation, to prepare users for the format. The format was chosen to be as concise as possible to allow 12 stories to be read consecutively without fatiguing the reader and affecting subsequent ratings.

The form of the stories relates to a struggle later in the project between the fabula and the discourse. All the components were created and put in order as the software generated the narrative, but a selection of JSON objects is unlikely to be considered a story. So this later stage of the project struggled with the difficulty of attempting to reconcile the fabula generated into a discourse that could be presented in a way that humans could enjoy (or not). With more time and thought the presentation would have been adjusted and perhaps incorporated into the user feedback more comprehensively. However the stories were introduced as outlines to manage expectations. So although this was an element that could undoubtedly have been polished, for efficiency and user experience, shorter and more quickly digestible narratives seemed appropriate.

Future Work

To a large extent the objectives set out in the introduction were met over the course of the project. However there are several areas in which the work could be developed further and potential alternate avenues of research that it could provide a foundation for.

An obvious continuation of the work could involve completing unfinished features such as allowing the mixing of components and consequences more freely. This would increase the variability of the output and allow for more combinations to be made with less reliance on pre-constructed fragments, also potentially increasing the likelihood for the unusual ideas and combinations that proved a fruitful creative source throughout the project.

If given more time, the project would also greatly benefit from the gathering of more user feedback. The feedback process could be refined and the story order randomised to reduce fatigue and potential bias. More feedback would help to see if the trends that started with a very small number of respondents hold over a larger group. As the scope of this project was small and focused on surrealist works, it would be fascinating to see if responses would differ if the genre were to change.

[3] It was suggested that this might be "sleight of hand" by one respondent. Hopefully this section proves there was nothing up the authors sleeve.

If the trends established by this project hold when a larger number of evaluations have been completed, it may be valuable to take a more fine grained approach to the research; perhaps investigating how constraints applied to one particular aspect of story such as action choice or character arc can affect the creativity of the resulting narratives.

Conclusions and Key Findings

When starting this project an early hypothesis was that the less constrained a narrative generation system was by rules or convention, the more potential for creativity existed. This hypothesis appears to be wrong, as the least constrained narratives were consistently chosen to demonstrate less creativity than the more constrained ones. However a potential sweet spot was found with minimal constraint applied to every aspect of narrative generation modelled except for character actions. The results show the areas in which constraint application appears are critical, but also highlight the freedom in other areas to relax what might be considered necessary impositions on the narrative generating agent.

A secondary hypothesis was that the existing crop of narrative generating software could be presented as a spectrum of constraint application to the problem of generating narratives. This is well supported by the feedback gathered from user evaluations, which shows a clear variance of response to narratives generated with differing levels of constraint, in a manner that supports the reading of the literature presented in section 2. This is further evidenced by respondents dislike for the more aleatory generation techniques and expressed preference for the more teleological, with particular focus on character arcs.

Another key finding was that the aesthetic tastes of evaluators are closely related to their assessment of creativity. So for artificial models of creativity to produce outputs which differ widely from established human standards and still be considered creative, the work should be explained or justified in some way by the creative agent.

Overall the data gathered shows promise for further investigation into the impact of constraints that may often be taken for granted when writing and evaluating narrative generating software and how their removal or adjustment may lead to more creative AI.

References

Benjamin. 2016. Sunspring http://www.thereforefilms.com/sunspring.html.

Boden, M. A. 1998. Creativity and artificial intelligence. *Artificial Intelligence* 103(1):347 – 356. Artificial Intelligence 40 years later.

Boden, M. 2004. *The Creative Mind: Myths and Mechanisms*. Routledge.

Colton, S., and Wiggins, G. A. 2012. Computational creativity: The final frontier? In *In Proceedings of 20th European Conference on Artificial Intelligence (ECAI) (pp. 21–26). Montpellier, France.*

Eck, D., and Schmidhuber, J. 2002. *Learning the Long-Term Structure of the Blues*. Berlin, Heidelberg: Springer Berlin Heidelberg. 284–289.

Jordanous, A., and Keller, B. 2016. Modelling creativity: Identifying key components through a corpus-based approach. *PLOS ONE* 11(10):1–27.

Jordanous, A. K. 2013. *Evaluating computational creativity: a standardised procedure for evaluating creative systems and its application*. Ph.D. Dissertation, University of Sussex.

Laclaustra, I. M.; Ledesma, J. L.; Méndez, G.; and Gervás, P. 2014. Kill the dragon and rescue the princess: Designing a plan-based multi-agent story generator. In *Proceedings of ICCC-2014, the 5th International Conference on Computational Creativity, Ljubljana, Slovenia, June 2014*.

León, C., and Gervás, P. 2008. Creative storytelling based on exploration and transformation of constraint rules. In *Proceedings of the 5th International Joint Workshop on Computational Creativity, Madrid, Spain*, 51–60.

Mueller, E. T., and Dyer, M. G. 1985. Daydreaming in humans and computers. In *In Proceedings of the 9th International Joint Conference on Artificial intelligence - Volume 1 (IJCAI'85), (pp. 278-280), Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.* Morgan Kaufmann Publishers Inc.

Mueller, E. T. 1990. *Daydreaming in Humans and Machines*. Norwood, NJ, USA: Ablex Publishing Corp.

Pérez y Pérez, R., and Sharples, M. 2004. Three computer-based models of storytelling: Brutus, minstrel and mexica. *Knowledge-Based Systems* 17(1):15 – 29.

Pérez y Pérez, R. 2014. The three layers evaluation model for computer-generated plots. In *Proceedings of ICCC-2014, the 5th International Conference on Computational Creativity, Ljubljana, Slovenia, June 2014*, 220–229.

Pérez y Pérez, R. 2015. A computer-based model for collaborative narrative generation. *Cognitive Systems Research* 36:30–48.

Pickering, T., and Jordanous, A. 2016. Applying narrative theory to aid unexpectedness in a story generation system. In Goel, A.; Jordanous, A.; and Pease, A., eds., *Eighth International Conference on Computational Creativity (ICCC'17) Atlanta, GA: ACC*, 213–220.

Sharp, O., and Goodwin, R. 2016. Machines making movies sunspring and project benjamin https://githubuniverse.com/2016/program/sessions/#machines-making-movies.

Sharples, M. 1996. An account of writing as creative design. In *The Science of writing. Hillsdale, NJ : Lawrence Erlbaum*. Routledge.

An Evaluation of Perceived Personality in Fictional Characters Generated by Affective Simulation

Leonid Berov and **Kai-Uwe Kühnberger**
Institute of Cognitive Science
University of Osnabrück
49076 Osnabrück, Germany
{lberov,kkuehnbe}@uos.de

Abstract

Readers can be drawn into a narrative through emotional engagement with its characters and their prospective fates. The type and extent of this engagement can be manipulated by providing the characters with distinct personalities. For this reason computational storytelling systems can benefit from explicitly representing personality. We present the results of an empirical study that evaluates whether the perceived personality of fictional characters created by our simulation-based narrative generator correlates with those computationally modeled. Motivated by the mimetic narrative theory of fictional minds the system models characters' action selection using an agent architecture grounded in a cognitive understanding of personality and affect. Results from our study support the claim that our system is capable of depicting narrative personality, that cognitive models are a viable approach to representing characters, and that a search-space of plot can be explored using character-personality as parameter. This can be taken to also provide functional evidence in support of the employed analytical narrative theory.

Introduction

Personality is an important property of fictional characters, which influences how readers engage with a work of narrative fiction: The personality of a character shapes how it is likely to act and react, which allows readers to form expectations about the possible future development of the plot. These expectations, in turn, allow readerly engagement in the forms of suspense and surprise (Sternberg 2001). Also, the moral evaluation of a character's personality is one of the grounds on which readers determine their disposition towards the character, a factor that is crucial for their emotional engagement with its fictional fate. A favourable disposition inclines the reader to a sympathetic response to a character's predicaments, whereas a negative disposition might even inhibit empathetic reactions (Schneider 2001; Eder 2006).

This means that computational storytelling systems can profit from explicitly modelling personality. Being able to generate plot while taking into consideration an explicit model of personality allows to create more plausible characters. That does not necessarily mean that characters always have to act consistent with their personality, but that the system needs to know when an action needs additional context in order to compensate for a deviation. At the same time, being able to represent characters more like readers perceive them means that a system is potentially more capable of deliberately manipulating readers emotional engagement with the generated stories.

Recently, the storytelling system Mask demonstrated how to express agreeable versus non-agreeable personalities through action choice using a planning-based approach (Bahamon and Young 2017). This corresponds to an external perspective on narrative, which understands characters as intentionally created effects (c.f. chapter 3, Currie 2010). We want to show that the perception of a distinct personality can also be generated using a different approach. Our algorithm is based on a multi-agent simulation system, where personality affects how characters process story-events and how they react to them (Berov 2017b). It does that by implementing a cognitively inspired model of affective reasoning. This amounts not only to a technological difference, but also to a change in narrative perspective. Characters are seen as non-actual individuals that are not mainly defined by actions or an intended effect, but rather through a (fictional) internal state: their beliefs, desires and emotions.

The present paper reports the outcome of an empirical study, in which we asked readers to rate the personality of fictional characters based on plots generated by our system. We investigated whether changes in the personality parameters of agents in our model correspond to changes in the perceived personality of the characters, as evaluated by readers using a personality questionnaire.

In the following sections, we will first present a short outline of work related to personality in narratology and psychology, which serves as grounding for our approach. Then, a high-level overview of the evaluated system will be presented to provide technical context. The experimental design will be introduced, and followed up with an evaluation of the obtained data, which gives significant evidence in support of the claim that our system is capable of modelling personality. A discussion of the results in the context of computational storytelling, and an outline of future work conclude the paper.

Related Work

Narratological Understanding of Personality

One of the earliest approaches to fictional characters classifies them into either *flat*, "constructed round [*sic*] a single idea or quality" ; or *round*, "capable of surprising in a convincing way" (Forster 1927, pp. 48, 55). On a closer look, both categories refer back to the concept of personality. The flat character is built around one marked and stable trait, which can be perceived as personality. The round character, on the other hand, can only be capable of surprising if readers first built up expectations about its action dispositions: it is by acting against this personality, that the character earns its roundness.

An explicit narratological treatment of personality can be found in the work of Palmer (2004). His main thesis is that readers reconstruct an understanding of characters from the text by attributing them with (fictional) minds that are the origin of the thoughts and actions reported on the surface. Crucially, he posits that these fictional minds, in most regards, work just like real minds. The theories of several other researchers point in the same direction: Zunshine (2006) observes, that readers apply their real-world trained Theory of Mind (ToM) in order to reliably derive mental states for characters, even when these states are not explicitly stated. It seems plausible to assume that in order for real-life ToM to be applicable to fictional minds they should work comparable to real minds. Schneider (2001) suggests a dynamic theory of how readers build mental models of characters, and observes that this process requires top-down knowledge processing, which imports real life knowledge into the models. This includes that human behavior is attributable to mental states, and a folk-psychological understanding of their interrelation. In a comparable vain, Ryan (1991) observes that all descriptions of story worlds are necessarily incomplete and suggests that readers follow a Principle of Minimal Departure in order to fill in blanks. This principle states that readers assume the nature of the story world to be comparable to the nature of the real world, unless explicitly stated otherwise. This, for Ryan, especially covers "our ideas of psychological laws" (p. 51).

Thus, Palmer grounds his understanding of fictional minds in the cognitive science discourse on real minds. He ascribes an especially important role in the working of fictional minds to "dispositions to behave in certain ways" (p. 108), something that he further describes by recounting Dennett's vivid image of "mind-ruts". While mainly speaking about action tendencies, Palmer also draws out personality's close connection to emotion tendencies (p.116). Being 'an angry person' implies a disposition to feel in certain ways and is as indicative a description of personality as e.g. being 'a violent person'. Naturally, emotions beget actions, and a separation of these dispositions is of mainly analytical value.

Following Palmer's cue, a more detailed understanding of dispositions and their quantification can be found in the real mind discipline of psychology.

Psychological Understanding of Personality

Quoting Child (1968), Eysenck (2004) describes personality as being comprised by "the more or less stable, internal factors that make one person's behaviour consistent from one time to another, and different from the behaviour other people would manifest in comparable situations" (p. 445). That is, personality can be used to distinguish individuals, describe their action tendencies and predict future behaviour. This definition captures what Palmer calls dispositions to behave in certain ways, and by that virtue further clarifies his terminology in relation to the aim of this paper.

While personality is one factor that contributes to a person's behavior, it is important to point out that the situation and context in which they find themselves has an at least equally important role (Eysenck 2004, p. 472). Thus, any computational model of personality-related reasoning should also model the influence of context, an observation that influenced the design of our character architecture.

A common approach to quantifying personality is to compare individuals based on a variety of distinct and limited traits. One of the most influential trait theories is the Big Five model (McCrae and John 1992) that proposes the five factors *openness to experience*, *conscientiousness*, *extraversion*, *agreeableness* and *neuroticism* (OCEAN) to comprehensively capture an individual's personality. Each factor is connected to several bi-polar scales, and high values on these scales are associated with typical adjectives that can be used to describe a person. For instance, trait extraversion is captured by the scales warmth, assertiveness or activity, and associated with adjectives like enthusiastic, outgoing or talkative.

Empirical evidence also confirm an interaction between Big Five personality traits and daily life affectivity. A recent study concerned with the nature of these interactions showed that trait neuroticism has a broad association with higher average levels, as well as a higher reactivity, of negative affect, while trait conscientiousness was associated with the opposite (Komulainen et al. 2014). This falls in line with Palmer's observations on affect and personality and is further evidence in favour of representing narrative personality in terms of the Big Five model.

The most common way to measure personality, according to Eysenck, is by way of self-report questionnaires, an approach that, unfortunately, is not readily available for the study of fictional characters. However, he points out that a reliable second way of personality assessment "is by ratings, in which observers provide information about other people's behaviour" (p. 457). In the context of narratives, readers are provided access to fictional characters through the narrator's description of action and thought and thus can be used as stand-in for actual observers. A recent questionnaire that has been successfully applied to self-report as well as observer ratings is the Berkeley BFI instrument (John, Donahue, and Kentle 1991; John, Naumann, and Soto 2008). Following the reasoning above, we assume that it can equally be used to rate the personality of fictional characters.

Computational Storytelling

Computational storytelling research is concerned with the study of algorithms that are capable of automatically generating fictional narratives (Gervás 2009). Two components of a narrative are distinguished. *Plot*: a content plane, which is a causally ordered series of events (what is told); and *discourse*: an expression plane, which is the linear representation of events in text form (Prince 2003). A common classification of plot generation systems is in *deliberative* and *simulation based* (Riedl and Young 2010). The former attempt to solve the problem of selecting a sequence of events based on a set of constraints, usually by means of a centralised reasoning algorithm. The latter employ decentralised, autonomous agents from whose interaction with each other and the environment a plot emerges. For a comprehensive overview of storytelling systems we refer to a survey by Kybartas and Bidarra (2017). Of these systems, to the best of our knowledge, only one other attempts to model narrative personality.

The Mask system (Bahamón, Barot, and Young 2015) employs a macro planner to solve the narrative planning problem, that is, to find a sequence of instantiated action schemata that transform the initial state of a story world into a state that satisfies a set of author goals. It thus falls mainly under the deliberative category. In contrast to classical planning problems, narrative planning problems commonly require that actions are only assigned to characters if they contribute to their intentions (Riedl and Young 2010). Mask models personality as an effect of choice. Whenever a character's intention can be achieved through different actions, the planning algorithm creates a branch for each alternative plan version and evaluates how well each branch portrays the personality trait that needs to be expressed for the character whose action created the branching point. The evaluation of consistency with a trait is defined for agreeableness. If a course of action supports other characters in achieving their intentions it is taken to indicate higher agreeableness, or the opposite if it prevents them (Bahamon and Young 2017). In the case that two branches result in contrasting evaluations, the consistent one is preserved as choice and included into the overall plan, while the other is not included in the plan but saved in order to ensure that it remains a possible path. Further steps are taken by the planer in order to prevent branches that are less contrasting than the choice/contrast pair.

As opposed to Mask, the system evaluated here falls into the simulation based category. It does not place character's action selection under the primacy of an overarching authorial plan, but on the contrary interprets the plot to emerge from the interaction of autonomous agents. Personality in our system is modeled, following Palmer's understanding of fictional minds, by means of a cognitively inspired affective reasoning algorithm.

The Character Architecture

The plot generation system evaluated in this paper is implemented as a multi-agent simulation system that is built around an affective agent architecture. Berov (2017b) presents how it was derived from the post-structuralist narrative theories of Ryan (1991) and Palmer (2004), and demonstrate how it can be used to explore a search space of possible plots. Here we will provide just a short overview of the employed agent architecture.

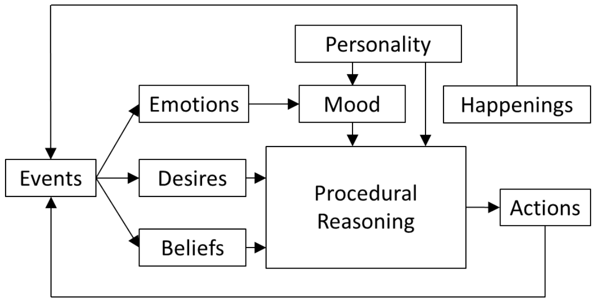


Figure 1: A high-level overview of the affective agent architecture.

The employed architecture extends the BDI agent architecture (Rao and Georgeff 1995) by adding the affective components emotion, mood and personality. The computational representation of these components and their interactions are modeled according to the Layered Model of Affect (Gebhard 2005), which is grounded in psychological research. The resulting reasoning cycle works as follows (also see fig 1).

Agents perceive their environment, and each perception is represented as an internal event that can change the agents belief base and/or activate a desire. An existing desire can be selected as an intention if at least one partial plan exists whose preconditions are met and that leads to the desired outcome. Each reasoning cycle, the agent selects one intention from the currently active ones in a round robin manner, and executes the next step in the associated plan. Since plans are partial, a plan step can either be an action or a sub-intention, which will be resolved to a plan only when all preceding plan steps have been executed. The resulting procedural reasoning algorithm is reactive since sub-intentions will be resolved according to the updated belief base on selection time.

Central to the affective reasoning components is personality. It is defined for each agent as a point in a five-dimensional space $[-1, 1]^5$, with the axes corresponding to the Big Five personality traits. Based on its personality, the agents default mood can be computed according to a formula derived by Mehrabian (1996, see their table 4).

Here, mood (i.e. medium-term affect) is represented as a point in a three-dimensional space $[-1, 1]^3$, with the axes *Pleasure*, *Arousal* and *Dominance* (PAD). An agent's current mood rarely stays at the computed default point. In any reasoning cycle in which the agent experiences emotions (i.e. short-term affect) a centroid is computed from all active emotions and the current mood moves further into the octant of this centroid. When no emotions are active the current mood slowly decays towards the default.

Emotions occur as the result of appraising internal events according to the OCC taxonomy (Ortony, Clore, and Collins 1990), which defines 22 distinct emotion types. In order to perform mood computations these emotions are translated into the PAD space according to Gebhard's mappings (see their table 2).

Thus, an agent's personality describes its inherent disposition to act and feel in certain ways, while its current mood aggregates a dynamic representation of context. The reasoning system uses both values as possible pre-conditions during plan selection. Additionally, the current mood can act as an elicitor of new desires (e.g. a mood low on P and high on A might elicit the desire to punish another agent).

The described character architecture is implemented as an extension of the Jason multi-agent simulation framework (Bordini, Hübner, and Wooldridge 2007) and available online[1]. It allows to manually model narrative systems in three steps:

- Implementing an environment in Java, which models the objective current state of the storyworld and implements the effects of agent actions,
- Implementing a common library of affective partial plans in AgentSpeak, which agents use during procedural reasoning, and
- Setting up one agent for each character, by defining their personality and potentially providing each with a set of unique beliefs and desires they hold on start-time.

A plot is generated automatically by executing the narrative system and is comprised of intentional events (character actions), non-intentional events (environment happenings) and internal events (private embedded narratives).[2]

Experimental Design

In order to evaluate whether the proposed system is capable of modeling the personality of fictional characters an empirical study was conducted. The study investigated if changing the personality parameters of a character in the computational model correlated with a significant co-directed change of the perceived personality of this character as judged by readers. The experiment presented below was designed to test the following null hypothesis: **"Changing a personality trait of a character in the model does not correlate with a change of readers' perceptions of the same personality trait in that character"**. In order to investigate whether a potential effect is diffused when multiple traits are changed at the same time two experimental conditions were tested.

To generate a control condition, the system introduced above was used to recreate the plot of the famous fairy tale "The Little Red Hen" (TLRH)[3]. This involved implementing a simple farm environment and four agents that take the roles of the respective characters: the hen and the three lazy animals (see table 1). Two experimental conditions were created by changing certain personality parameters of the protagonist:

1. condition E: the extraversion trait was lowered ($E = -0.3$),
2. condition NA: the neuroticism trait was lowered ($N = -1$) while the agreeableness trait was raised ($A = 0.7$),

which resulted in plots that differed from the basic condition by at least one action executed by the protagonist.[4]

	HEN	ANIMALS
O	0	0
C	1	-1
E	0.7	0
A	0.3	-0.7
N	0.15	-0.8

Table 1: Personality parameters used to model the four characters of TLRH (control condition).

Plots are represented in the system by directed graphs that contain the actions executed, the events perceived, and the emotions experienced by each character. No textualization module exists so far that could be used to translate these graphs into story text. To create a textual form of the two experimental conditions a collaborator was recruited. She was presented with the system-generated graph for the control condition B as well as the text of the original fairy tale, and then asked to transform the graphs of condition E and NA into story texts based on the provided example pairing. To avoid the unconscious introduction of biased text the collaborator was not informed about the hypothesis of the experiment or the provenance of the graphs. The crafted story text was identical to the original tale whenever the same situations were described, and only differed in the context of different actions taken by the protagonist.

An online survey platform was used to carry out the study. 40 participants were recruited from the University of Osnabrück through e-mail and social media. A within-subject design was selected in order to reduce interpersonal differences in the data and allow meaningful results with the available number of participants. Each participant was presented with the texts of all three conditions, and each text was instantly followed by a personality survey about the protagonist. The personality survey used the 44 statements from the BFI instrument, and asked participants to indicate how much they perceived these statements as applicable to the protagonist (e.g. "Little Red Hen is helpful and unselfish with others"). Participants could provide answers to each statement using a 5-item Likert-scale from 1 (strongly disagree) to 5 (strongly agree).

In order to avoid introducing a systematic bias due to carry-over effects from the first story/question pair to following conditions, presentation order of the three conditions was randomized between participants. Feedback from a pretrial suggested that participants found it hard to mentally separate the protagonist (Little Red Hen) of the later conditions from preceding ones. To facilitate the task, and avoid

[1] https://github.com/cartisan/plotmas

[2] These are the major components of plot in Ryan's (1991) narrative framework. Embedded narratives capture each character's subjective interpretation of the unfolding plot and are necessary to explain narrative phenomena like deception or failed plans.

[3] www.home.uos.de/leberov/tlrh.htm

[4] www.home.uos.de/leberov/tlrh_versions.htm

	control	condition E	condition NA
O	3.12 +/- 0.44	2.87 +/- 0.53*	3.11 +/- 0.46
C	4.50 +/- 0.37	4.26 +/- 0.56*	4.42 +/- 0.41
E	3.79 +/- 0.48	2.56 +/- 0.65****	3.72 +/- 0.54
A	3.02 +/- 0.72	2.82 +/- 0.52	4.58 +/- 0.33****
N	2.39 +/- 0.66	2.58 +/- 0.60	1.86 +/- 0.60****

Table 2: Survey results: perceived personality (mean +/- std) of the protagonists of the three conditions. Asterisks indicate significant difference with control condition.
*: $P \leq 0.05$, ****: $P \leq 0.0001$.

this non-systematic carry over effect, the protagonists of the three conditions were additionally assigned different names. The names were randomly selected from the top ten most common female names in the US over the last 100 years[5], in order to avoid name-related biasing. The resulting protagonist names were Little Red Hen Linda (control condition), Little Red Hen Mary (condition E) and Little Red Hen Susan (condition NA). As will be discussed in the next section these measures proved to be sufficient, as we interpret the data to show no significant effect of condition order on personality judgement.

The collected data for each participant includes demographic data, the order in which conditions were presented, and the answers from three BFI inventories relating to the three conditions. The inventory data was post-processed according to the instructions provided by the instruments authors (no ipsatization was applied). While the collected inventory data was discrete, the resulting average scores are continuous values in the range from 1.0 to 5.0. The results provide the five average personality trait scores of the protagonist as perceived by the readers for the three experimental conditions (see table 2).

Data Evaluation

The gathered experimental data allows answering the research question—formulated in the null hypothesis above—by checking for significant effects in the affected traits. This can be done by analysing whether changing a character's personality trait in the model correlates with a co-directional change in perceived personality. It is also of interest to check whether the employed personality model has the desirable property of being orthogonal, that is, whether a change in one trait also modifies the perception of other traits. It also allows validating the employed within-subject design by checking for interaction effects between subsequently presented stories.

A Mauchly Test for all obtained trait ratings showed that the sphericity assumption was violated in the data. Therefore, in the following all repeated measure ANOVA results are reported with a Greenhouse-Geisser correction.

[5]Based on `https://www.ssa.gov/oact/babynames/decades/century.html`

Condition-order effect

It can be assumed that no interaction effects arise between subsequently presented conditions if a condition's protagonist's perceived personality does not change significantly in dependence of the position in which the condition was presented to the participant. To determine this, the five personality trait ratings for the protagonist of the control condition were compared between three groups: participants who read the story first ($N = 16$), ones that read it second ($N = 11$), and last ($N = 13$)[6]. A single factor independent measure ANOVA was executed for each personality trait. No significant between-group differences were found in the O,C,E and A traits. A significant between-group difference was found in the N trait (at $P = 0.025$). A post-hoc pairwise Tukey HSD showed significant differences between the third group and the other two groups, however, no significant difference between the first and second groups ($\mu_{N_{1st}} = 2.17$ vs. $\mu_{N_{2nd}} = 2.24$ vs. $\mu_{N_{3rd}} = 2.79$).

The last result requires further analysis, since it indicates that presentation order affects personality rating only for N traits and when the story is presented in third position. An ANOVA tests the assumption that all samples were drawn from the same population. Thus, the pairwise post-hoc tests suggest that the first and the second samples, as well as the second and the third samples, were drawn from the same population, however, the first and the third samples originate from different populations. Being drawn from the same population is a transitive property, which indicates inconsistent results. In the present case, five statistical tests where executed on the same data, of which one found a significant effect at an $\alpha = 5\%$ level. Considering that ANOVA tests are not corrected for multiple comparisons the probability of finding a false positive in this setting is around 20%. Taking these observations together we interpret the last finding to be a random sampling effect, and not an effect of condition order.

We conclude that no interaction effects take place between subsequent conditions, which validates the choice of a within-subject design. The same analysis could be conducted for the other two stories but was left out due to time constraints.

Perception of modified traits

A single factor repeated measure ANOVA shows a highly significant difference (at $P = 5.85 \cdot 10^{-14}$) in the perceived extraversion between groups. A pairwise post-hoc comparison between the control condition ($\mu_{E_{control}} = 3.78$) and condition E ($\mu_{E_E} = 2.56$), where the parameter was lowered, demonstrates that the effect shows in the correct direction.

Single factor repeated measure ANOVAs show a highly significant difference between the perceived neuroticism ($P = 5.48 \cdot 10^{-5}$) and agreeableness ($P = 1.29 \cdot 10^{-17}$) between groups. A pairwise post-hoc comparison between the control condition ($\mu_{N_{control}} = 2.39$, $\mu_{A_{control}} = 3.02$)

[6]The unequal distribution of participants is due to a technical limitation of the employed survey software which only allows a randomized presentation order instead of a true counterbalancing.

and condition NA ($\mu_{N_{NA}} = 1.86$, $\mu_{A_{NA}} = 4.58$), where N was lowered while A was raised, demonstrates that the effects show in the correct direction.

We conclude that the null hypothesis can be rejected, that is, changing parameters in the model correlates with a change in perceived personality along the modified traits, and the effect shows the right direction. For a statistical summary see tables 3, 4 and 5 in the appendix.

Perception of non-modified traits

The data is less conclusive on the account of trait orthogonality. Single factor repeated measure ANOVAs (see tables 3 through 7 in the appendix) in combination with post-hoc pairwise comparisons show the following results, which indicate orthogonality:

- there is no significant difference between the O, C and E traits of the control condition and condition NA ($P_O = 1.00$, $P_C = 0.43$, $P_E = 0.79$),
- there is no significant difference between the N and A traits of the control condition and condition E ($P_N = 0.29$, $P_E = 0.23$).

At the same time, the following results indicate no orthogonality:

- there is a significant difference between the C and O traits of the control condition and condition E, although neither C nor O were changed there ($P_C = 0.0013$, $P_O = 0.0063$).

First, it can be observed that the unintended effects are only present in condition E. This is a surprising finding, since we expected condition NA to have a higher interaction potential due to the higher number of changed personality parameters in the model (2 versus 1). We propose two possible, non mutually exclusive interpretations: (1) in the employed computational personality model the traits N and A are orthogonal to the other traits, whereas E is not orthogonal to at least O and C, (2) in the story-domain model a change in E trait leads to a more prominent change in behavior than a change in N and A, which propagates to stronger changes in perceived personality. The first interpretation is supported by the fact that several studies show evidence for significant intercorrelations between at least some of the five traits (Eysenck 2004, p. 468). The second interpretation is supported by the observation that condition E differs from control by four missing actions (three times the hen doesn't ask for help and one time she doesn't offer to share the bread) whereas condition NA differs by only one action (the hen shares the bread instead of eating it alone).

Second, it can be observed that, while the unintended effects are significant, they are several magnitudes weaker than the intended ones. This is a desirable property since it potentially allows to counteract unintended interactions by coordinated changes in the dependent personality traits. Whether it is practically possible to negate interactions in such a way remains to be ascertained empirically.

Discussion

The experimental findings presented above allow to reject the null hypothesis. This means that changes in agent's personality parameters using our system correlate with co-directed changes in the perceived personality of fictional characters represented by these agents. Several conclusions can be drawn from this finding.

Conclusions

1. The employed affective agent architecture is capable of modeling personality.
2. Simulation-based approaches can be used to manipulate plot, based on character's personalities.
3. Cognitively inspired personality theories can be used to model the literary effect of personality.
4. A within-subject design is viable for the study of personality in the context of computational storytelling.

It is worth discussing the conclusions in detail. (1) In Berov (2017a) the claim that the affective agent architecture is modeling personality was based only on the theoretical argument that it implements a psychological model of personality. However, a psychological model is necessarily an abstraction from the observed phenomenon, and an implementation of a model is necessarily a simplification. Our results indicate that these various transformations preserved the phenomenon they modeled. This means we now also have functional arguments to support the above claim: the architecture models personality, because it creates the appearance of personality. (2) Bahamon and Young (2017) have demonstrated that a planning-based approach can be used to model narrative personality. We show, that this is also viable in a simulation-based approach. Furthermore, by showing that the parameters of the proposed system indeed represent personality, we can claim with more certitude that it models the narrative theory foundations we set out to implement. Arguing from the reverse, we can now also give support to Palmer's (sometimes contested) claim that fictional minds work like real minds, by demonstrating that this assumption can indeed lead to the generation of coherent plot. (3) Scepticism is sometimes expressed in the community towards the approach of using cognitively inspired theories to model fictional characters. One common criticism is that it is implausible to assume that writers use scientific theories of mentation to devise their characters, but rather rely on their folk-psychological understanding. This difference might lead to problematic deviations in character modeling. Especially, folk psychology allows to generate predictions even for extreme cases, while scientific theories should work more reliably in average situations (Pablo Gervás, personal communication). Our experiment suggests that cognitive personality persists under narrativization, that is, personalities that can be described using the Big Five can be created in the reader's impression. We would like to point out that it was observed that individuals with personality disorders have extreme scores on the five traits (Eysenck 2004, p. 467), suggesting that also non-average cases should be, in principle, viable. Indeed, some of the hen's behavior already modeled in our case study leaned towards the quixotic. But even if more archetypal or pathological cases were not representable using our system it still should remain useful as a baseline measure. As explained above, to realize any surprising, unexpected behavior a system first needs to infer the plausible, consistent reaction. This opens up interesting avenues for further exploration: Pizzi and Cavazza (2007) model affect in their system Madame Bovary using a set of literary feelings outlined by Flaubert himself. Would it be possible to achieve comparable effects in a system using cognitive emotions? It might at least address the problems reported by the authors when attempting to port their system to a BDI framework (Peinado, Cavazza, and Pizzi 2008). (4) Previous empirical studies of character personality in the context of computational storytellers relied on a between-subject design to rule out interaction effects. This comes at the cost of requiring a high number of participants. Our results indicate that instead also a within-subject design can be used, if executed with certain precautions like counterbalancing and character individuation. This has the advantage of requiring significantly less volunteers by reducing inter-individual variability.

Future Work

The presented model of personality is sufficient to create flat characters that behave in idiosyncratic and predictable ways. To create round characters, however, it is necessary to enable them to behave in ways that stand in contrast with their personality. To some extent this is possible with the present approach because, apart from personality, it also takes into consideration the character's context during action selection. An additional way to enable surprising behavior would be to allow character's personality to change during the course of a narrative. Technically this is possible because an agent's personality is represented as a point in the OCEAN space and can be updated at any time. The challenge is to find ways to automatically relate events that occur during a simulation run with traits that might be affected by them, and to identify when a change should take place. A continuous change in trait values based on recurring situations could represent the character's gradual adaptation to their environment, while a discrete change over a comparably large Δ_X could represent traumatic change due to incisive events. Character development and learning are an important aspect of more character-oriented genre like e.g. the bildungsroman and by that virtue pose an inherent interest for storytelling systems.

A commonly identified drawback of simulation-based storytelling systems in the context of computational creativity is their tendency to produce quotidian interactions due to a lack of narrative regulation. This means that in order to turn the presented system from a computational model of narrative into a computational model of storytelling, mechanisms need to be implemented that guide plot generation based on prospective interestingness. One such notion of interestingness is Ryan's (1991) tellability (see chapter 8), which attempts to capture the aesthetics of plot based on purely structural properties. Berov (2017c) suggests an approach towards the computational modeling of this measure, which would allow an automatic post-hoc evaluation of the aesthetic quality of a particular plot instance. Armed with such a measure, a storyteller based on our character architecture should be capable of exploring a plot space by iteratively executing a simulation, measuring its tellability, and changing the constraints that influence subsequent iterations. In Boden's (1990) terms this would mean an alternation between exploratory and transformational creative processes. At the moment, our system would allow to perform such an automatic exploration of plot space based on personality parameters. Future work includes expanding this set to include parameters like environment-controlled events (happenings) and properties of the environment itself.

Acknowledgments

The first author is grateful for support for this work provided by an Alexander von Humboldt Ph.D. fellowship funded by an Anneliese Maier-Forschungspreis awarded to Mark Turner. We also like to thank Elena Livchits for performing the textualization of the two stories for the experimental conditions.

References

Bahamon, J. C., and Young, R. M. 2017. An empirical evaluation of a generative method for the expression of personality traits through action choice. In *Proceedings of AIIDE-17*, 144–150. Snowbird, Utah, USA: AAAI Press.

Bahamón, J. C.; Barot, C.; and Young, R. M. 2015. A goal-based model of personality for planning-based narrative generation. In *AAAI*, 4142–4143.

Berov, L. 2017a. Character focused narrative models for computational storytelling. In *Proceedings of AIIDE-17*, 277–279. Snowbird, Utah, USA: AAAI Press.

Berov, L. 2017b. Steering plot through personality and affect: an extended BDI model of fictional characters. In *Proceedings of KI 2017*, 293–299. Cham: Springer.

Berov, L. 2017c. Towards a computational measure of plot tellability. In *Proceedings of INT-10*, 169–175. Snowbird, Utah, USA: AAAI Press.

Boden, M. 1990. *The Creative Mind: Myths and Mechanisms*. London: Weidenfeld and Nicholson.

Bordini, R. H.; Hübner, J. F.; and Wooldridge, M. 2007. *Programming Multi-agent Systems in AgentSpeak Using Jason*, volume 8. John Wiley & Sons.

Child, I. L. 1968. Personality in culture. *Handbook of personality theory and research* 82–145.

Currie, G. 2010. *Narratives and narrators: A philosophy of stories*. Oxford University Press.

Eder, J. 2006. Ways of being close to characters. *Film Studies* 8(1):68–80.

Eysenck, M. W. 2004. *Psychology: An International Perspective*. East Sussex: Psychology Press.

Forster, E. M. 1927. *Aspects of the Novel*. New York: RosettaBooks LLC, 1st electronic (2010) edition.

Gebhard, P. 2005. ALMA: a layered model of affect. In *Proceedings of AAMAS-05*, 29–36. ACM.

Gervás, P. 2009. Computational approaches to storytelling and creativity. *AI Magazine* 30(3):49–62.

John, O. P.; Donahue, E. M.; and Kentle, R. L. 1991. The big five inventory–versions 4a and 54. Berkeley, CA: University of California, Berkeley, Institute of Personality and Social Research.

John, O. P.; Naumann, L. P.; and Soto, C. J. 2008. Paradigm shift to the integrative big five trait taxonomy: History, measurement, and conceptual issues. In John, O. P.; Robins, R. W.; and Pervin, L. A., eds., *Handbook of Personality: Theory and Research*. New York, NY: Guilford Press. 114–158.

Komulainen, E.; Meskanen, K.; Lipsanen, J.; Lahti, J. M.; Jylhä, P.; Melartin, T.; Wichers, M.; Isometsä, E.; and Ekelund, J. 2014. The effect of personality on daily life emotional processes. *PLoS One* 9(10):e110907.

Kybartas, B., and Bidarra, R. 2017. A survey on story generation techniques for authoring computational narratives. *IEEE TCIAIG*.

McCrae, R. R., and John, O. P. 1992. An introduction to the five-factor model and its applications. *J. Pers* 60(2):175–215.

Mehrabian, A. 1996. Analysis of the big-five personality factors in terms of the pad temperament model. *Aust J Psychol* 48(2):86–92.

Ortony, A.; Clore, G. L.; and Collins, A. 1990. *The Cognitive Structure of Emotions*. Cambridge University Press.

Palmer, A. 2004. *Fictional Minds*. U of Nebraska Press.

Peinado, F.; Cavazza, M.; and Pizzi, D. 2008. Revisiting character-based affective storytelling under a narrative BDI framework. In *ICIDS-2008*, 83–88. Springer.

Pizzi, D., and Cavazza, M. 2007. Affective storytelling based on characters' feelings. In *Intelligent Narrative Technologies: Papers from the AAAI Fall Symposium*, 111–118.

Prince, G. 2003. *A Dictionary of Narratology*. U of Nebraska Press.

Rao, A. S., and Georgeff, M. P. 1995. BDI agents: from theory to practice. In *Proceedings of the ICMAS-95*, 312–319.

Riedl, M. O., and Young, R. M. 2010. Narrative planning: Balancing plot and character. *JAIR* 39:217–268.

Ryan, M.-L. 1991. *Possible Worlds, Artificial Intelligence, and Narrative Theory*. Indiana University Press.

Schneider, R. 2001. Toward a cognitive theory of literary character: The dynamics of mental-model construction. *Style* 35(4):607–639.

Sternberg, M. 2001. How narrativity makes a difference. *Narrative* 9(2):115–122.

Zunshine, L. 2006. *Why We Read Fiction: Theory of Mind and the Novel*. Ohio State University Press.

Appendix

Repeated measure ANOVA results for the five traits and the groups: control, condition E, condition NA. The Greenhouse Geisser correction is reported because the sphericity assumption is violated in the data.

ANOVA	**SS**	**df**	**MS**	**F**	**P value**
Subjects	19.54	39	0.50	2.30	0.00090
Groups	37.90	2	18.95	86.00	1.37 E-20
Error	16.99	78	0.22		

Greenhouse Geisser	**SS**	**df**	**MS**	**F**	**P value**
Groups	37.90	1.83	20.67	86.10	5.85 E-14
Error	16.99	71.51	0.24		

Table 3: Results for trait **extraversion**.

ANOVA	**SS**	**df**	**MS**	**F**	**P value**
Subjects	21.01	39	0.54	1.78	0.06
Groups	11.22	2	5.61	18.50	2.66 E-07
Error	23.66	78	0.30		

Greenhouse Geisser	**SS**	**df**	**MS**	**F**	**P value**
Groups	11.22	1.79	6.26	18.50	5.48 E-05
Error	23.66	69.92	0.34		

Table 4: Results for trait **neuroticism**.

ANOVA	**SS**	**df**	**MS**	**F**	**P value**
Subjects	21.01	39	0.54	1.78	0.06
Groups	11.22	2	5.61	18.50	2.66 E-07
Error	23.66	78	0.30		

Greenhouse Geisser	**SS**	**df**	**MS**	**F**	**P value**
Groups	11.22	1.79	6.26	18.50	5.48 E-05
Error	23.66	69.92	0.34		

Table 5: Results for trait **agreeableness**.

ANOVA	**SS**	**df**	**MS**	**F**	**P value**
Subjects	17.32	39	0.44	4.97	9.12 E-10
Groups	1.24	2	0.62	6.91	0.00172
Error	6.97	78	0.09		

Greenhouse and Geisser	**SS**	**df**	**MS**	**F**	**P value**
Groups	1.24	1.48	0.83	6.91	0.01099
Error	6.97	57.78	0.12		

Table 6: Results for trait **conscientiousness**.

ANOVA	**SS**	**df**	**MS**	**F**	**P value**
Subjects	16.83	39	0.43	3.51	1.19 E-06
Groups	1.60	2	0.80	6.51	0.00242
Error	9.59	78	0.12		

Greenhouse Geisser	**SS**	**df**	**MS**	**F**	**P value**
Groups	1.60	1.83	0.88	6.51	0.01285
Error	9.59	71.28	0.13		

Table 7: Results for trait **openness**.

Redesigning Computationally Creative Systems For Continuous Creation

Michael Cook[1] and **Simon Colton**[1, 2]
[1]The MetaMakers Institute, Falmouth University, UK
[2]Computational Creativity Group, Goldsmiths, University of London, UK
www.metamakersinstitute.com ccg.doc.gold.ac.uk

Abstract

Most systems developed for Computational Creativity projects are run for short periods of time, which provides enough space to be creative, but limits the long-term growth and development of software both internally and in the wider world. In this paper, we describe the notion of *continuous creativity* by describing how ANGELINA, an automated game designer, was rebuilt to be an 'always-on' system. We review the history of ANGELINA and contrast this new approach with earlier versions. We introduce the term *presence* to describe the impact a computationally creative system has on its environment, and vice versa, and discuss how continuous creativity can contribute to a system's presence, providing greater creative independence, opportunities for framing, and space for the system to grow.

Introduction

Automated game design is a frontier challenge for Computational Creativity research. Composing music, writing stories, conceptualising visual aesthetics, inventing systems of rules – designing a videogame involves solving many distinct creative problems, and ensuring all of those solutions pull together towards the same ultimate goal. (Liapis, Yannakakis, and Togelius 2014) describes videogames as the 'killer app' for Computational Creativity, as they offer not only a variety of creative challenges, but also additional problems of co-operation and integration between creative tasks and potentially between different creative individuals.

ANGELINA is an automated game design project which has been developed, over several iterations, to explore ideas relating to Computational Creativity. The project's aim has been twofold: to solve the hard, technical problems of developing software capable of designing games automatically; and to investigate the social and cultural aspects of game design and try and understand how, if at all, an AI system can take on a role in this space. As a result, our work on ANGELINA encompasses studies of evolutionary computation and code synthesis, as well as user studies, exhibitions, and evaluating the cultural impact of the software. Each version of ANGELINA is designed to focus on a particular subproblem within automated game design, but shares a common core structure and engineering approach (Cook 2015).

Rebuilding and reassessing the system over the course of many years allowed us to refocus the project on new problems, as well as embrace emerging technology and new platforms. However, it also led to a lack of continuity between the different versions of ANGELINA, and had a negative impact on the perception of the system as a long-term creative entity. Observers found it hard to make sense of a system which changed so often, and ANGELINA had little persistence outside of a single creative act. Reflecting on these issues, we designed and built a new version of ANGELINA, with an emphasis on long-term creative existence. We called this new approach *continuous automated game design*, or CAGD, but more generally it expresses an approach to computational creativity we are calling *continuous creativity*.

In this paper, we discuss the process of redesigning ANGELINA in the continuous creativity paradigm. We begin by giving an overview of the history of the project and discuss its limitations, introducing the concept of *presence* to describe the aggregated long-term legacy of a computationally creative system. We then outline the new structure of ANGELINA, explaining the changes necessitated by a shift to continuous creativity. Finally, we place this new version of ANGELINA in the context of related automated game design research, and then discuss future work. The rest of the paper is organised as follows: in *Background* we cover the history of ANGELINA as a project; in *Presence In CC Systems* we critically reflect on the project and introduce the notion of *presence*; in *Designing For Presence* we describe the new ANGELINA, focusing on how the structure of the system has changed in response to the demands of continuous creativity; in *Discussion and Future Work* we discuss the potential problems and new directions exposed by continuous creativity; in *Related Work* we place ANGELINA in the context of automated game design research.

Background

ANGELINA is an automated game design system, and has been in development in some form since 2011, first presented at ICCC in 2013 (Cook, Colton, and Gow 2013). Although version numbering is somewhat obscured by project forks and anonymised systems, there are five distinct historical versions of ANGELINA, each with a different focus, but their broad structure remains the same across most versions. In this section we give a brief overview of the structure of the software, and then dissect some of its shortcomings. We also provide an overview of related work in terms of automated game design and computationally creative systems. Throughout this paper we occasionally refer to a specific version of ANGELINA with a subscript like this: ANGELINA_1. For a discussion of ANGELINA's various versions to date, see (Cook, Colton, and Gow 2017).

Each version of ANGELINA can be thought of as running through three distinct phases: predesign, design and postdesign, as described below.

Predesign

The system begins in a **predesign** phase. For early versions of the software this predesign phase was almost nonexistent, simply loading in parameters, but for later versions of the software this is where it would lay the foundation for the design problem it was about to tackle. ANGELINA_3 read online newspaper articles to decide on an inspiring article to make a game about, and would then search for online media like images and sound effects to use in its game design. In ANGELINA_5, the system was provided with a short phrase or single word before running, and broke the input down using online concept and word association databases.

Predesign was a flexible space which we could use to add in new functionality during the setup for the system. Because the design phase which follows it cannot be interrupted until it is finished, the predesign phase was the only way for the system to make creative decisions before work on the game began. Any expansion to ANGELINA's creative capacity typically had to fit into this phase somehow: for example, the system was later given the ability to scrape social media to assess how people felt about a particular notable figure, which then became a parameter that influenced other media searches. This easily fitted in to the predesign phase, because it ultimately only affected the media that would be fed into the design phase. However, we were unable to expand the system in more complex ways – responding to a serendipitous discovery while designing the game by going back and searching for more information online, for instance.

Design

The **design** phase is the largest and most important part of the system. All major versions of ANGELINA employ the same technique in this process, namely *cooperative coevolution* (Potter and Jong 1994). Unlike a normal evolutionary system, where a single population of solutions is evaluated and recombined until termination, a cooperative coevolutionary system is composed of several separate evolutionary subsystems solving their own part of a larger problem, in this case designing a game. When an evolutionary subsystem evaluates its population, instead of evaluating it in isolation it uses high-ranking exemplars from the other subsystems to synthesise a larger artefact, in this case a game, and then evaluates that larger artefact instead. Thus, the fitness of a population member is not just based on local evaluation, but on evaluation in the context of a larger solution.

Different versions of ANGELINA were built to break the game design process down into different kinds of subsystem – ANGELINA_1 had Level Design, Layout Design and Ruleset Design, for instance. Our intention was to replicate the way a small game developer might distribute the creative task of game development, where level designers, musicians, writers and so on would work independently, but share their work together to evaluate their progress towards a common goal. Another desirable feature this technique had was being omnidirectional. In many generative systems there is a clear line of steps that the software always moves through, in the same order. A cooperative coevolutionary system does not work in this way, because all parts of the artefact are being evolved simultaneously. If, for example, the Level Design subsystem struck on a particularly good design, this could influence the fitness landscape of the other subsystems.

Postdesign

After the design phase has concluded and the game is not going to be changed further, we enter the **postdesign** phase. The most important part of this phase is compiling the game – in many cases, ANGELINA would not create an executable when it was finished. ANGELINA_2 modified Actionscript files and then ran a compiler to produce a game, while ANGELINA_5 required the manual moving and arrangement of files so it could be compiled into a finished binary executable. Although this may sound like a minor aspect of the system compared to coevolutionary systems and creative evaluation, the requirement for the intervention of a person is a major weakness, both from the perspective of the perception of observers, and the autonomy and independence of the system. We believe that it is crucial that the system can release its work on its own, so it can control how and when it disseminates its work. This is a big part of 'closing the loop' creatively – allowing the system to decide when it is finished.

Another aspect of postdesign was the preparation and compilation of framing information (Charnley, Pease, and Colton 2014). Across all versions which employed framing this took the form of textual commentaries, which ANGELINA would construct using templates filled with data about decisions made by the system. Because of the dense nature of the design phase, the framing information never referenced the development of the game itself – instead, it discussed intentions and motivations, and the origins of the media used in the games. The main reason for this was that it was hard to convey meaningful things about the design phase because players never knew what it was or what took place in it, they only ever saw the finished game. Almost everyone we spoke to asked about the design process itself, highlighting how little we communicated about it.

Presence In CC Systems

ANGELINA began as an abstract system with little emphasis on creative decision-making, and evolved over time to take into account issues like real-world context, self-evaluation, and framing. However, this did not change the fundamental structure of the system, or the AI techniques it used to achieve its goals, and this caused problems as the project developed. In this section, we highlight some of the most common issues we identified in the design and execution of ANGELINA, and then introduce a common thread which ties them together: the concept of *presence* in a computationally creative system.

Opacity Of The Design Phase

A working definition of Computational Creativity makes reference to 'unbiased observers' who assess software as being creative or not based on how it behaves (Colton and Wiggins 2012). The most important aspect of ANGELINA's process, where the game is actually designed, was not only impossible to observe, but would also be impossible for many to understand even if they could observe it (Cook, Colton, and Gow 2013). As such, they can only guess at how ANGELINA develops games in the Design Phase.

We found this was a particular problem for ANGELINA, because observers were unable to distinguish work done by ANGELINA from work done by us in building the system. For example, for ANGELINA$_3$ we built a template game for the system to modify. This meant that aspects of the game like how the camera moved, the player's appearance or the control scheme were all out of ANGELINA's control. Players frequently attributed this to ANGELINA, however, because they didn't know enough about the design process to know what the system was actually responsible for.

Short Term Impact

The second major problem, which we believe may have been exacerbated by the opaque nature of the design phase, is that people were unimpressed or confused by the higher-level structure of the system. We believe that because they were unable to assess ANGELINA effectively by examining its design work, they instead looked to other aspects of the system for evidence of creative autonomy, such as how the software operates not when creating a single artefact, but across its entire lifespan. Common questions asked by both journalists and the general public included:

- Can the system learn new things?
- How does the system decide when to make a new game?
- How many games has it made?
- Can it play other people's games and learn from them?

None of these questions refer to the act of designing a game: instead they touch on the long-term growth of the system; whether the system has creative independence; what the system's legacy is; and whether it can engage with the existing culture of videogames. It speaks to a higher-level thinking about AI, one that is willing to accept that the system can perform certain tasks, but now wants to know what those tasks are in aid of, and whether they are in the context of a wider environment.

Previous versions of ANGELINA lacked good answers to these questions. We decided when to run ANGELINA, and what it should make a game about. The act of creation left almost no long-term impact on ANGELINA – in a rare case, ANGELINA$_3$ would remember past topics and respond slightly differently if it came across them again. It could not engage with other creators (the closest we came was having other creators engage with it, when its game jam entries were judged by its peers). It did not develop over time, and it had no long-term goals – it was designed to run as a blank slate, create something, and then stop. Our use of co-operative co-evolution also hampered us, because during co-evolution every aspect of the game's design was constantly in flux, *and* constantly dependent on every other aspect of the design, which made it hard to extend the design phase or add systems that modified or adapted the design process.

Different Creative Modalities

Besides the opacity of the design phase and the lack of long-term structure, there was also a technical problem we encountered when designing previous iterations of ANGELINA, namely that we found it hard to balance high-level design work and fine-grained discovery work. ANGELINA$_4$ used metaprogramming to invent new game mechanics, but in order to do so, it would exhaustively simulate an existing game with specific objective functions. This was intensive work just to discover a new game mechanic in a fairly stable search space. On the opposite end, ANGELINA$_5$ used parameterised game mechanics and simple level design algorithms so it could rapidly prototype and test games, allowing it to explore a higher-level design space more quickly. Combining these in a single system would be difficult. There would be no way to have both activities working simultaneously in a cooperative coevolutionary system because a small change in the high-level design would seriously disrupt the low-level search for new game mechanics.

At the same time, discovering game mechanics, or any other kind of detailed design knowledge, felt like something that should happen outside of an ordinary game creation loop. Yet there was no obvious place to put this, because ANGELINA was only run when we intended for it to design a game, and there was no clear plan for how newly-discovered design knowledge would be fed into ANGELINA's normal game design process. Discovering new game mechanics produces no immediately consumable creative artefact, yet seems like an important part of the system's growth. Who should decide what kind of task it undertakes, or when it undertakes them? It felt difficult to build a system in the way we had been, while enabling all these different motivations and modalities for creativity.

Product, Process and Presence

(Jordanous 2015) notes that "traditionally within computational creativity the focus has been on... [a] system's Product or its Processes" – by which they mean the artefacts produced by software, and the way in which those artefacts were made. In reflecting on our work on ANGELINA, we propose a third element to this line of thinking, which we call *Presence*. Presence is the impact a computationally creative system has on its environment, and the impact the environment has on that system in return. It accumulates over time, and encompasses both tangible things (such as a system's knowledge of its past work) and intangible things (the perception the public has of the system). To put these three elements in context: *product* relates to a single artefact, at the moment it is consumed by an observer; *process* relates to the means by which that artefact came into being; *presence* relates to the impact of the system's history and environment on the process being undertaken, and the impact the resulting product will have on the future of the system and its environment. Presence is not merely the sum total of the system's output – it also includes things such as how the system influences and is influenced by its peers and critics; how the system relates to and is perceived by its audience; how the system sets and achieves goals for itself; how it learns and grows through creating.

At the time of writing, many systems in Computational Creativity have a presence, but it is almost entirely sustained by the involvement of the system's designers. For example, ANGELINA's presence is sustained by talks given by the authors about the system; by a website of projects that is manually maintained by the authors; by public events like entering game jams or releasing games that are chosen by the authors. It is not detrimental to the system's creativity to have people contribute to a system's presence, indeed it may never be possible to fully separate a system's creator from that system's legacy. However, the software must also have some responsibility in creating and managing its own presence, as a step towards us handing over creative responsibility to a system, and enabling software to have creative autonomy not just over what they make, but on their place in the wider world, and any creative communities they may exist within the context of.

In redesigning ANGELINA, our intentions were to find a way to increase this sense of presence in the system. We aimed to do so not simply by adding features to the software, but by designing the structure of ANGELINA in such a way that it would not need to be rebuilt as often as previous versions, and in a way that encouraged future additions to the system to preserve and expand the system's presence. In the next section we describe how we went about doing this.

Designing For Presence In ANGELINA

We designed the latest version of the system, ANGELINA$_6$, to take into account the conclusions of our project review, and the identification of presence as a lacking element in the system to date. The result is a design which we hope will not only produce better games, and frame them in richer and more compelling ways, but also a system with more control over its own presence, and a better foundation on which to build new features and do more research in the future, without rebuilding the system again. In this section, we provide a high-level overview of some of the system's most important features, before discussing lessons learned from this process in the following section.

Overview

ANGELINA$_6$ maintains a database of active game designs that it is working on, each of which has a metadata file which tracks important statistics about the project and tasks that need to be completed. When the system has completed a task, it checks this database and selects a project that has active tasks and is not on hold. After ANGELINA$_6$ loads the project by parsing the game's project file (written in a domain-specific game description language, described below), it selects a pending task on the project's to-do list and passes control to a module designed to complete that particular task. When ANGELINA$_6$ has completed its current task, it will modify the project file, updating the game if the task was completed successfully, and making notes in the metadata file for future work. If the game is ready to release or needs to be abandoned, it may perform additional steps here, otherwise it files the game back in the database and begins the cycle of selecting a new game to work on.

Continuous Creativity

One of the most important changes in this version of ANGELINA$_6$ is that the system does not have a defined start or end point. Instead of being turned on, creating a game, and then stopping, ANGELINA$_6$ is designed to constantly cycle through a database of active game projects with associated lists of pending tasks. It can also choose to start a new project to add to this list, or declare a project as abandoned or released to remove it from the list. Theoretically speaking, ANGELINA$_6$ can now run indefinitely, moving between creative tasks and producing games forever. In practice, we do not actually run ANGELINA$_6$ perpetually for reasons of energy conservation and hardware strain, but the system resumes exactly what it was last doing when it is restarted. The decision to design ANGELINA$_6$ as a continuous system was one of the earliest decisions we made when redesigning the software, and it forms the core of what we call *continuous creativity*, a way of building software that we describe in detail in (Cook 2017).

Making the system continuous is the most important design decision made in the new version of ANGELINA$_6$. A continuous structure gives us the ability to have the system change the order in which it performs creative tasks, or even change which creative projects it works on. It also raises questions about how these systems should be upgraded, how often – if at all – a system should be reset, and how the data within them should be structured. By forcing ourselves to commit to the notion that this software is always working, always existing in the world, we change our relationship with the software as creators, and put the long-term presence of the software above short-term research goals. The software is now in control of what it does and when it does it, it decides when to start work on something and when to change to something else or stop entirely. This shifts the relationship between the public, the system and us as its programmers, and puts more emphasis on the system's autonomy and independence.

Task-Driven Design

Prior versions of ANGELINA$_6$ used cooperative coevolution to simultaneously design all aspects of a game together. The main advantage we perceived this as having was that all aspects of the design could be solved simultaneously, and therefore any part of the design could 'lead' and influence other parts. However, this approach came with many drawbacks, including a higher complexity for observers and overcorrection between subsystems. The new ANGELINA$_6$ eschews this approach, and instead breaks up each part of the design into its own separate task – thus, when ANGELINA$_6$ is designing a level now it is only designing a level, and nothing else is happening at the same time. We provide ANGELINA$_6$ with a catalogue of tasks for different purposes, which can be parameterised to specialise a task to a particular game or phase of development. Current tasks include designing rulesets, sketching level concepts, designing levels, and assigning art and colour schemes to the game's content. Each task employs its own process for completing its work: for example, level design uses a mix of evolutionary design and MCTS for testing levels (Browne et al. 2012), while ruleset design uses abductive reasoning and answer set programming (Gelfond and Lifschitz 1988).

The most immediate benefit from this is clarity and transparency: it's now simple to express to observers what the system is doing at any given time, because it is only ever doing one thing. We also gain a new kind of nonlinearity to the way the system works, despite giving up the simultaneous nature of the coevolutionary approach. Currently, ANGELINA$_6$ is given a loose structure in how it designs games: design a ruleset; experiment with level design to confirm the ruleset's potential; design several larger levels to fit the game; release the game. However, in the future as ANGELINA$_6$'s task catalogue expands, we plan to give the system the autonomy to dynamically change its task queues to fit a particular game. For example, it might discover that it cannot design many interesting levels, and so schedules a task to extend the ruleset and make it more complex. After doing this, it will schedule further level design tasks, as well as another task to evaluate the older levels and confirm they still work in the context of the new ruleset.

This is all possible because the task system is entirely modular and written with clear interfaces that ANGELINA$_6$ can use. For example, the Level Design task can be customised to change aspects such as the size of the level, the complexity of the desired solution, and the depth with which to search for solutions. This means ANGELINA$_6$ can easily adjust the same modular task to accommodate exploratory design, simple level design, and deep ruleset exploration. In the future, we hope this will lead to ANGELINA$_6$ having a lot of autonomy over how it works, and provide it with opportunities to refine its work and go back and improve on tasks that are already completed.

Longer Design Cycles

Continuous work shifts the emphasis of the software away from producing a single game and towards growth as a game designer over many creative acts – in other words, it emphasises *presence* over *process*. An individual game project is now just a section in the long-term existence of the system, rather than the target outcome of running the system. This also removes the need to generate a game by a deadline – previously we would want ANGELINA$_6$ to produce a game relatively quickly because the system could not save its work, and thus had to create a game in a single execution. A continuous system doesn't need to work in this way, and so we are using this as an opportunity to build a system that spends weeks producing an artefact rather than hours.

One of the reasons people were fascinated by how long it took ANGELINA$_6$ to produce a game is that AI, particularly creative AI, can seem mysterious to the general public. Even though ANGELINA$_6$'s games were not blockbuster-quality, the idea that it only took four or six hours to make one seemed impressive. One of the benefits of changing the timeframe of ANGELINA$_6$ is that it shifts its work from being on the scale of software to being on the scale of humans. This isn't just a perceptual benefit, however. Working more slowly means we have more opportunities for observers to engage with the process – ANGELINA$_6$ can tweet about a game idea it has had, and blog about the development process over multiple weeks, culminating in the release of the game. This allows people to see development and growth during creation, not just after the fact as has been the case before. This is a new approach to framing for the project.

This also opens up opportunities for ANGELINA$_6$ to work with people more directly. Game developers frequently collaborate with others to complete a game, and also send their game to playtesters to get feedback. Up until now, ANGELINA$_6$'s short timescale has meant that it has had to play its own games, and acquire pre-existing art and music from online sources. But working over weeks means ANGELINA$_6$ can send out games to testers and wait for feedback, or send commissions to artists and musicians and wait for responses. These are exciting opportunities for research into human-software collaboration, and longer timescales make it feel like a natural part of the process.

Custom Engine & Description Language

Past versions of ANGELINA$_6$ used game templates designed by hand which they then modified and exported. This restricted the systems more, but made it easier to build them in the first place, and much easier to disseminate the finished games which was always a key objective. This version of ANGELINA$_6$, much like ANGELINA$_5$, is built in the Unity game development environment, but unlike ANGELINA$_5$ its output is a text file, not a Unity project. This text file contains the entire game described in a custom description language we have made, inspired by VGDL (Schaul 2013) and Puzzlescript (Lavelle 2014). The text files act like game cartridges or ROMs, in that they are fed into another application which we have created, which interprets the language and runs the game. The interpreter uses a custom game engine which we built in Unity, meaning that both ANGELINA$_6$ and the interpreter use exactly the same code to run games.

The immediate advantage to this approach is that it makes it easier to distribute games, and easier for ANGELINA$_6$ to release them. Almost all prior versions of ANGELINA$_6$ needed some manual work by a person to compile and distribute its games, but now it can upload that text file to game marketplaces, or send them via email. Using a description language also has additional benefits though, primarily that it allows other people to easily write games that can be interpreted by ANGELINA$_6$. This means that for the first time ANGELINA$_6$ can play games designed by other people and learn design knowledge from them, or evaluate them and give feedback to the designer. We intend to explore this in future work and investigate how ANGELINA$_6$ can work

```
{
    "trigger": "OVERLAP enemy playerpiece",
    "code": [
        "DESTROY $2",
        "SFX punch2",
    ]
},
{
    "trigger": "ENDTURN",
    "code": [
        "DO_AI_HUNT enemy playerpiece"
    ]
}
```

Figure 1: A code snippet from a game description.

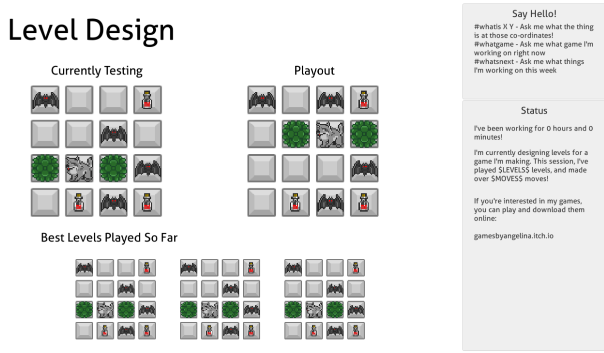


Figure 2: A screenshot of ANGELINA's workspace, on the Level Design view.

with and learn from other people.

Our decision to use a custom language rather than an existing one is partly down to other languages not quite fitting our needs – Puzzlescript is quite abstract for software generation, and we felt the VGDL was too prescriptive. The most important reason, however, is that we wanted a language which was flexible enough to enable ANGELINA$_6$ to extend it in the future. Figure 1 shows part of a game description, to illustrate this. The part shown defines two rules in the game, each structured as a trigger condition followed by a list of things that happen when the condition is met. The top rule says that when an enemy overlaps with the player, the player piece is destroyed. The second rule says that when a turn ends, enemies move towards the player.

We've designed the description language so that ANGELINA$_6$ can engage with it at different levels depending on the kind of design work it is doing. At the highest level, it treats the entire code in Figure 1 as a single game concept that it can add into a game without modification (it adds enemies which chase the player and kill them). ANGELINA$_6$ has a catalogue of these mechanics that it can use to rapidly develop games with concepts that are known to be useful. It can also create its own rules, using the language to design triggers and lists of effects. This is a lower-level action that would probably be performed outside of a game design, in a prototyping phase where it experiments with new game ideas. When it finds useful or interesting mechanics, it can add them into its catalogue to use later in higher-level design tasks. Finally, it can work at an even lower-level, and use metaprogramming and code generation techniques to add new keywords (like `DESTROY`) to the language. We aim to extend our previous work in mechanic discovery to do this (Cook et al. 2013). These new keywords could then be used in low-level mechanic design, and ultimately filter up into high-level catalogues of mechanics. Being able to work at different levels fits in with the overall philosophy of continuous creation and growth.

Discussion and Future Work

The notion of presence is relevant to all areas of Computational Creativity, and we believe that many of the engineering decisions made in the latest version of ANGELINA$_6$ also have wide applicability to most of the domains that Computational Creativity has been applied to. As with *process*, there is no step-by-step guide to emphasising *presence* in a system. Nevertheless, we believe the following three features of ANGELINA$_6$ were useful in helping drive us towards building a system with more presence:

- **Continuous** – The system has no beginning or end, and seamlessly moves between tasks and projects, recording its progress, starting and stopping projects as it sees fit.
- **Modular** – The system selects from several tasks in order to advance a project, and tackles a single activity at a time.
- **Long-Term** – The system is built with long timeframes in mind; a single project can take a long time to produce, and a single project is less significant than the impact it has on the system's creative development.

These features help us think about the system beyond a single creative act, and about how the system will change over time, what interfaces it presents to the outside world, and the ways in which it can be extended in the future.

Continuous creativity and exploring the notion of presence opens up a lot of future work – investigating how to motivate long-term systems, how to build more complex framing profiles using historical data, as well as raising questions about how a system should be tested, how it should be reset, and whether unofficial execution of the software, such as during development, constitutes as part of the system's official 'history'. For now, we identify two key areas of future work that we intend to pursue:

Long Timescale Visualisation

ANGELINA$_6$'s creative process is now much more accessible, because it only works on one task at a time, and organises itself in a way that is perhaps closer to how people organise large creative tasks (such as maintaining lists of short-term goals, or evaluating progress as a project moves towards completion). This means that people can now watch ANGELINA$_6$ as it creates, something which has been trialled before by software such as The Painting Fool (Colton and Ventura 2014). However, the continuous nature of the software changes the tenor of this experience, as people can now observe the system over longer periods of time, which allows them to notice changes in a particular artefact being created, as well as growth in the system itself.

We plan to explore this by having ANGELINA$_6$ livestream its creative process on online streaming sites such as Twitch.tv. This will allow people to watch ANGELINA$_6$ as it works, and we have designed a visual frontend to the software that tries to represent ANGELINA$_6$'s creative activities in a way that faithfully represents the underlying algorithmic activity. Figure 2 shows one of the design screens. At the time of writing we have completed some trial streams, and also had ANGELINA$_6$ exhibit at a major games expo, which we hope to report on in a future publication.

Richer Framing

Because ANGELINA$_6$ records its progress in such detail, including maintaining lists of tasks and projects, version history for its games and notes on the success or failure of tasks, the system has a huge amount of data at its disposal about the creative process. This is somewhat necessitated by the continuous, modular nature of the system – since it has to be able to suspend projects and transfer data between modules, it has to keep meticulous records and copious metadata about each creative project it starts. This, combined with the slower, long-term nature of the system, opens up powerful new ways for the system to frame its work to observers.

We also have opportunities for framing *during the creative process* which is not something we believe has been attempted before in Computational Creativity. Because the creative process aims to last days or even weeks, we can have the system comment and reflect on its process while it is still working on a project. This provides even greater opportunities than before to have a system remark on changes in direction, leaps in progress, and crucial decisions. Even though many computationally creative systems exhibit these properties, they are rarely highlighted during short, intense generative processes. Continuously creative systems, however, provide natural points between tasks where a system can reassess its work and identify next steps, or make notes about the process for later framing.

In tandem with our livestream experiments, we are also developing ANGELINA$_6$ to engage in more active forms of framing, by allowing viewers of the livestream to ask ANGELINA$_6$ questions using a limited set of phrases the system understands. This allows viewers to retrieve framing information from the system dynamically, at different stages of development. Examples of these questions include: asking what project the system is working on currently; asking what other tasks they have to do next; or asking what a specific game piece does in the game being worked on. We plan to study the impact of this active framing on the perception of the software as creative; we anticipate it will have a positive impact and help connect observers more closely to the creative process during creation, rather than only allowing engagement after the fact.

Related Work

Automated game design is a growing area of study, and is beginning to fork into a set of subproblems that share a common core. One of these is the generation of test cases for general game playing – unseen games, or games designed to specifically test a particular area, would help research into general game playing, and also has benefits for certain kinds of competitive human play. Research that closely links general game playing to game design, such as (Khalifa et al. 2017) and (Bontrager et al. 2016), are forging a link between these problem domains, as is the emergence of a game design track in the General Video Game AI competition (Liebana et al. 2016). This challenge-first approach can be traced back to work by (Togelius and Schmidhuber 2008), for example, who designed rulesets for games based on how hard they were to learn.

Another application area is automated design as support for other game designers. The Sentient Sketchbook (Liapis, Yannakakis, and Togelius 2013) is a tool that assists in the level design process, with an innovative interface that helps sort and visualise important information and opportunities to the user. In a similar vein, (Shaker, Shaker, and Togelius 2013) present Ropossum, an interactive level design tool for physics-driven games like Cut The Rope. While these tools don't try to take on the entire game design process, they use very similar techniques and show how AI tools can assist in a variety of different game design contexts.

Similarly, work by Osborn proposes that this broader goal of 'discovering game design knowledge' should be one of the field's objectives (Osborn, Summerville, and Mateas 2017), something that echoes a paper by Smith, one of the earliest game design papers at ICCC, which proposed the concept of a machine that discovered game design knowledge through experimentation (Smith and Mateas 2011). We are already seeing work aimed at discovering or translating game design knowledge, for example through machine vision and interpretation (Guzdial and Riedl 2016) (Guzdial, Li, and Riedl 2017), or reasoning about game design knowledge using formal methods (Martens et al. 2016).

Many other systems exist simply to further the broader goal of building software that can design games. Sometimes this is focused on a narrow genre, such as Barros et al's work on mystery puzzle games (Barros, Liapis, and Togelius 2016), while others attempt broader systems that target a less complex but also less fixed structure, such as the Game-o-Matic, possibly the most successful automated game design project to date (Treanor et al. 2012). These systems often tackle the hard problems of cultural knowledge, too, such as Nelson and Mateas' system which built simple games from plain text descriptions (Nelson and Mateas 2008).

Conclusions

In this paper, we described a new version of ANGELINA, rebuilt to reflect our changing ideas about computational creativity and automated game design. We introduced the notion of *presence* in computationally creative systems, to complement well-established notions of *process* and *product*. We showed that past versions of ANGELINA lacked presence, and how redesigning a new version of the software to be continuously creative helped guide us towards a design that can take more responsibility for its own presence and long-term growth. Finally, we laid out our immediate next steps for ANGELINA$_6$, and some next steps for those looking to incorporate these ideas into their own systems too.

Acknowledgments

This work is funded by EC FP7 grant 621403 (ERA Chair: Games Research Opportunities). The authors wish to thank the reviewers for their feedback on the paper; the Max Planck Institute for Software Systems; the automated game design community for many years of discussion that shaped ANGELINA up to this point; and Chris Donlan, for asking the right questions.

References

Barros, G. A. B.; Liapis, A.; and Togelius, J. 2016. Murder mystery generation from open data. In *Proceedings of the International Conference on Computational Creativity*.

Bontrager, P.; Khalifa, A.; Mendes, A.; and Togelius, J. 2016. Matching games and algorithms for general video game playing. In *Proceedings of the Conference on Artificial Intelligence in Interactive Digital Entertainment*.

Browne, C.; Powley, E.; Whitehouse, D.; Lucas, S.; Cowling, P. I.; Tavener, S.; Perez, D.; Samothrakis, S.; Colton, S.; and et al. 2012. A survey of monte carlo tree search methods. *IEEE TRANSACTIONS ON COMPUTATIONAL INTELLIGENCE AND AI*.

Charnley, J.; Pease, A.; and Colton, S. 2014. On the notion of framing in computational creativity.

Colton, S., and Ventura, D. 2014. You can't know my mind: A festival of computational creativity. In *Proceedings of the International Conference on Computational Creativity*.

Colton, S., and Wiggins, G. A. 2012. Computational creativity: The final frontier? In *ECAI*, Frontiers in Artificial Intelligence and Applications.

Cook, M.; Colton, S.; Raad, A.; and Gow, J. 2013. Mechanic miner: Reflection-driven game mechanic discovery and level design. In *Proceedings of the EVOGames Workshop, Applications of Evolutionary Computation Conference*.

Cook, M.; Colton, S.; and Gow, J. 2013. Nobody's a critic: On the evaluation of creative code generators. In *Proceedings of the International Conference on Computational Creativity*.

Cook, M.; Colton, S.; and Gow, J. 2017. The ANGELINA videogame design system - part I. *IEEE Trans. Comput. Intellig. and AI in Games* 9(2):192–203.

Cook, M. 2015. *Cooperative Coevolution For Computational Creativity: A Case Study In Videogame Design*. Ph.D. Dissertation, Imperial College, London.

Cook, M. 2017. A vision for continuous automated game design. In *Proceedings of the Experimental AI and Games Workshop at AIIDE*.

Gelfond, M., and Lifschitz, V. 1988. The stable model semantics for logic programming. 1070–1080. MIT Press.

Guzdial, M., and Riedl, M. O. 2016. Game level generation from gameplay videos. In *Proceedings of the Conference on Artificial Intelligence in Interactive Digital Entertainment*.

Guzdial, M.; Li, B.; and Riedl, M. O. 2017. Game engine learning from video. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI*.

Jordanous, A. 2015. Four pppperspectives on computational creativity. In *AISB 2015 Symposium on Computational Creativity*.

Khalifa, A.; Green, M. C.; Liebana, D. P.; and Togelius, J. 2017. General video game rule generation. In *IEEE Conference on Computational Intelligence and Games*.

Lavelle, S. 2014. Puzzlescript. http://www.puzzlescript.net/.

Liapis, A.; Yannakakis, G. N.; and Togelius, J. 2013. Sentient sketchbook: Computer-aided game level authoring. In *Proceedings of ACM Conference on Foundations of Digital Games*.

Liapis, A.; Yannakakis, G. N.; and Togelius, J. 2014. Computational game creativity. In *Proceedings of the International Conference on Computational Creativity*.

Liebana, D. P.; Samothrakis, S.; Togelius, J.; Schaul, T.; and Lucas, S. M. 2016. General video game AI: competition, challenges and opportunities. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Martens, C.; Summerville, A.; Mateas, M.; Osborn, J.; Harmon, S.; Wardrip-Fruin, N.; and Jhala, A. 2016. Proceduralist readings, procedurally. In *Proceedings of the Experimental AI and Games Workshop at AIIDE*.

Nelson, M. J., and Mateas, M. 2008. An interactive game-design assistant. In *Proceedings of the 13th International Conference on Intelligent User Interfaces*.

Osborn, J. C.; Summerville, A.; and Mateas, M. 2017. Automated game design learning. In *Proceedings of the IEEE Conference on Computational Intelligence in Games*.

Potter, M. A., and Jong, K. A. D. 1994. A cooperative coevolutionary approach to function optimization. In *Proceedings of the International Conference on Evolutionary Computation*.

Schaul, T. 2013. A video game description language for model-based or interactive learning. In *Proceedings of the IEEE Conference on Computational Intelligence in Games*. IEEE Press.

Shaker, N.; Shaker, M.; and Togelius, J. 2013. Ropossum: An authoring tool for designing, optimizing and solving cut the rope levels. In *Proceedings of the Conference on Artificial Intelligence in Interactive Digital Entertainment*.

Smith, A. M., and Mateas, M. 2011. Knowledge-level creativity in game design. In *Proceedings of the International Conference on Computational Creativity*.

Togelius, J., and Schmidhuber, J. 2008. An experiment in automatic game design. In *Proceedings of the IEEE Conference on Computational Intelligence in Games*.

Treanor, M.; Schweizer, B.; Bogost, I.; and Mateas, M. 2012. The micro-rhetorics of game-o-matic. In *Proceedings of the Foundations of Digital Games Conference*. ACM.

Generalization across Contexts in Unsupervised Computational Creativity

Debarun Bhattacharjya[1], Dharmashankar Subramanian[1], and Lav R. Varshney[2]
[1]Research AI, IBM T. J. Watson Research Center
[2]Electrical and Computer Engineering, University of Illinois at Urbana-Champaign
{debarunb,dharmash}@us.ibm.com, varshney@illinois.edu

Abstract

Autonomous computational creativity systems must not only have the ability to generate artifacts, but also to select the best ones on the basis of some assessment of quality (and novelty). Such quality functions are typically directly encoded using domain knowledge or learned through supervised learning algorithms using labeled training data. Here we introduce the notion of unsupervised computational creativity; we specifically consider the possibility of unsupervised assessment for a given context by generalizing artifact relationships learned across all contexts. A particular approach that uses a knowledge graph for generalizing rules from an inspiration set of artifacts is demonstrated through a detailed example of computational creativity for causal associations in civic life, drawing on an event dataset from political science. Such a system may be used by analysts to help imagine future worlds.

Introduction

Computational creativity (CC) systems are intended to help generate artifacts that are deemed to be creative by their users or experts in the domain. Although creative value is subjective and often difficult to pin down, there is consensus in the literature that creative artifacts should be novel as well as valuable or useful (Boden 1990; Mayer 1999). Since novelty can be viewed as one of potentially many attributes of creative value (Bhattacharjya 2016), we use the term *quality* to refer to all non-novelty related aspects of creative artifacts (Ritchie 2001; Pease, Winterstein, and Colton 2001).

Clearly, a crucial requirement for any autonomous CC system is the ability to evaluate the creative value of artifacts, particularly their quality. A popular approach to evaluating quality is through the use of extensive domain-specific knowledge. For instance, the IBM Chef Watson system exploits knowledge from hedonic psychophysics around chemical compositions and flavor profiles of individual ingredients (and their combining rules) to evaluate the potential pleasantness of recipes (Varshney et al. 2013). An alternate approach to evaluating quality is to learn it from assessments of artifacts provided by other agents, typically humans. Examples include the PIERRE system for stew recipes that uses human-specified ratings for complete recipes (Morris et al. 2012) and the DARCI system for images that receives feedback from humans (Norton, Heath, and Ventura 2010).

In the aforementioned systems, the mechanism for evaluating quality is explicitly specified; we refer to this as *supervised* computational creativity, whether achieved by supervised learning on complete artifacts or by encoding properties of components and combining rules for combinatorial creativity. In contrast, we posit that quality can be inferred in numerous ways without such explicit knowledge in *unsupervised* computational creativity, using an *inspiration set* (also known as *inspiring set* (Ritchie 2001)) of artifacts and potentially additional knowledge that does not pertain directly to the quality of artifacts. In this paper, we describe a specific data-driven framework that uses a knowledge graph in addition to the inspiration set. We illustrate the approach through a novel application where we first build an inspiration set of cause-effect pairs from a political event dataset and then use these to generate creative cause-effect pairs of events occurring in a country.

The key high-level idea behind our approach is that of *generalization*, i.e. when there is information about artifacts from various contexts, one might be able to learn across all these contexts to estimate a proxy measure for quality. In particular, if there are patterns that seem to be widely prevalent, they could indicate characteristics of high-quality artifacts; the underlying assumption is that widespread prevalence hints at potential usefulness. An artifact could then be contextually creative if it is contextually novel, i.e. original/surprising for a particular context, but also adheres to generally prevalent patterns. While the notion of generalization in computational creativity has been described previously, e.g. Ventura (2016), here we make the connection to quality evaluation – an essential module for any CC system.

Although supervision has played a dominant role in practical CC systems and will likely continue to do so in the future, we believe that the supervised/unsupervised distinction is useful for the field to consider. For one, unsupervised computational creativity forges a path to pursuing abstract conceptual work, thereby enabling ideas and formulations that could be useful across application domains. Further, it explicitly extends the role of machine learning in computational creativity, cf. Guzdial and Riedl (2018). In a recent review, Toivonen and Gross (2015) discuss the role of data mining and machine learning in computational creativity:

as far as evaluation of quality is concerned, they focus on supervised techniques. In contrast, we consider the use of unsupervised learning techniques. We begin by expanding upon the distinction around supervision.

Supervision in Computational Creativity

There are two fundamental approaches to supervision in computational creativity: 1) to use domain knowledge to map combinations of components of artifacts to measures of quality, and 2) to learn such quality functions from labels such as user ratings, typically through supervised machine learning techniques.

Quality Functions from Domain Knowledge

Computational creativity applications span diverse application domains such as the visual and culinary arts, music, poetry/narrative generation, and mathematical and scientific discovery. It is not surprising that many successful CC systems rely heavily on knowledge specific to their domain of application, pertaining to the quality of an artifact. This enables the formulation of models that explicitly relate specific combinations of components of artifacts to measures of quality that are appropriate for the application domain. We illustrate this with a couple of examples, one from the culinary arts and one from the sciences.

Chef Watson. The Chef Watson system is designed to produce novel and flavorful culinary recipes (Varshney et al. 2013; Pinel, Varshney, and Bhattacharjya 2015). Bayesian surprise, a domain-agnostic information-theoretic notion of novelty, is used together with a combination of two domain-specific measures of flavor quality. The first measure pertains to olfactory pleasantness, drawn from hedonic psychophysics theory and computed from molecular properties of individual flavor compounds present in individual ingredients, together with a combining rule to predict the percept of complete dishes from the individual compound properties. The second is a notion of flavor pairing, drawn from the network science of culinary practice, and was originally validated using positive examples of recipes from large corpora. It is also computed using the flavor compound composition of ingredients. As can be noted, the evaluation of quality in this system requires access to detailed hedonic psychophysics and chemoinformatics data.

HAMB (Heuristic Autonomous Model Builder). As a knowledge discovery system that has been deployed in biological sciences applications like protein crystallization (Livingston 2001; Buchanan and Livingston 2004), HAMB is different from other CC systems in that the form of the eventual creative product is different from that of artifacts in the inspiration set. HAMB receives an empirical dataset as input and returns a set of discovery items. These items are varied; the most prevalent kind is a conditional rule that classifies features/attributes based on other features in the dataset (ex: if f_1 and f_2 then f_3 with p-value p). The quality of a discovery item in HAMB is its interestingness, quantified using the system builders' expertise around the knowledge discovery process. For example, for a rule or a rule set, it is measured through standard performance metrics for classification such as precision and recall, p-value, etc. HAMB is a prime example of how artifact quality in a CC system is modeled using rich knowledge about the domain, in this case that of rule induction for knowledge discovery.

Learning from Quality Labels

An alternate approach to supervision in computational creativity is through the availability of what we refer to as *quality labels*. These labels are indications from sources such as previously acquired datasets or real-time human assessments with explicit information about the quality of artifacts. When such labels are available, they can be used to learn one or more quality functions, typically using supervised machine learning methods. Once again, we provide specific examples, from the visual and culinary arts.

NEvAr (Neuro Evolutionary Art). In the NEvAr tool, populations of images are generated through an interactive evolutionary process (Machado and Cardoso 2002). Like previous evolutionary art tools, the underlying representation of an image is a tree of mathematical functional operators applied to x-y coordinates of pixels in the image. In NEvAr, the user guides the highly interactive process by selecting individual images and providing a fitness score. Images have a default fitness value of 0 but the user could choose a small set of preferred images and provide a score greater than 0, typically 1 to 3. This approach is typical of CC systems involving genetic algorithms, where human-assisted supervision is performed in real-time.

PIERRE (Pseudo-Intelligent Evolutionary Real-time Recipe Engine). PIERRE is a recipe generation system for crock pot recipes, i.e. soups, stews and chilis, that uses online recipes as an inspiration set (Morris et al. 2012). Like NEvAr, PIERRE uses a genetic algorithm for generation. Crossover is performed by splitting the two parent recipes into two sub-lists each and merging these, and mutation includes changes to ingredient amounts as well as ingredient replacements, additions, and deletions. Supervision in PIERRE occurs through user ratings of recipes which are also available in their repository. A multi-layer perceptron is used to perform a regression that connects an input layer of real-valued amounts of ingredient groups to a real-valued output node of rating (between 0 and 1) through a 16-node hidden layer. The system builders also added negative examples by assigning a 0 rating to randomly generated recipes.

We note that CC systems can be varied and complex in their architecture as well as in the extent and timing of human involvement; this can make it difficult to strictly categorize or contain the mechanism of supervision. CC systems that use case-based reasoning, for instance, could potentially rely on various forms of supervision. An example is poetry generation using COLIBRI (Diaz-Agudo, Gervas, and Gonzalez-Calero 2002) where supervision is achieved by finding the nearest case but also by word substitution using domain knowledge about poetry such as part-of-speech, rhyme, and syllable matching.

Another interesting supervised system is The Painting Fool (Colton 2012) which uses a pipeline of techniques to modify initial domain-specific quality function knowledge. Colton (2008) describes an approach that produces scenes similar to downtown Manhattan where the fitness of the size, shape, color, and location of rectangle placeholders are hand-crafted; an evolutionary model then invents new fitness functions. A practical complication is that a system may work in different modes, perhaps with different types of supervision. NEvAr, for instance, uses quality labels when in interactive evolutionary mode, and author-provided domain knowledge about the aesthetic appeal of an image (based on compression metrics) when in fully automated mode.

A fundamental issue with supervised computational creativity approaches is that it is difficult to transfer quality evaluation modules from one application domain to another. Another issue is that when a system is tied to pre-specified notions of quality, it could miss out on productive regions of the conceptual space of artifacts (Wiggins 2006). Unsupervised techniques could potentially open up the playing field around domain-agnostic quality evaluation in CC systems.

Unsupervised Computational Creativity

In unsupervised computational creativity, one must attempt to create without the help of an explicit quality function. An approach that is popular is to take an inspiration set of unlabeled positive examples from the domain, learn models to mimic the style and then make modifications of the learned representation. A classic example is the work of David Cope in music creativity, which models the styles of great composers like Bach and Mozart, and then creates new examples of music ranging from single-instrument arrangements to full symphonies (Cope 1996). This approach also allows mixing of two or more different styles.

A modern reincarnation of this approach uses deep neural networks and generative adversarial networks in creative domains, building on their recent successes in machine learning. An example is the work of the Google Magenta project[1] with applications in music and visual art. In certain aspects, this approach to creativity can be limiting. When modifications to the learned representation are minor, resulting artifacts can be perceived to be too close to those in the inspiration set; from an artistic perspective, some have therefore criticized the results as pastiche. When modifications are major, the resulting artifacts may be of low quality, particularly since these systems do not typically have a means to judge their creations. To help avoid such issues, it could be beneficial to use proxies for quality for evaluating artifacts. As Colton and Wiggins (2012) write: “A poet with no critical ability to judge its own work ... is no poet at all”.

Unsupervised computational creativity is clearly a challenging endeavor and necessarily requires making assumptions. This is analogous to machine learning, where unsupervised methods such as clustering implicitly assume that objects that are similar in the feature space are more likely to belong in similar clusters.

[1] https://magenta.tensorflow.org

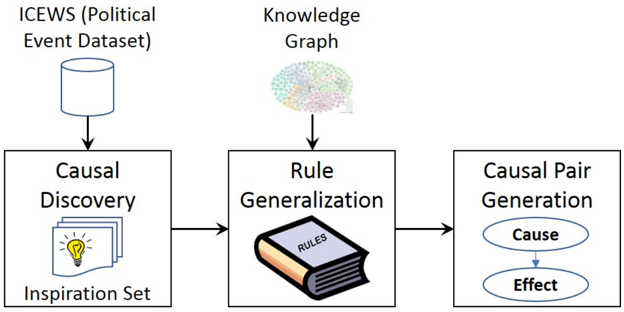


Figure 1: Workflow in the causal association application.

One can further note that due to the absence of any sort of evaluation, a final selection step is often carried out by humans. This is true for Cope’s work but also for many supervised systems, including Harold Cohen’s AARON for visual art (McCorduck 1990).

Contextual Computational Creativity

We refer to the type of computational creativity formulation underlying the application in the next section as *contextual computational creativity*, where it is assumed that there is access to an inspiration set with artifacts pooled together from various contexts. Formally, the dataset is $\mathcal{I} = \{(z_i, c_i)\}_{i=1}^{M}$ where z_i is the ith artifact and c_i is the context of the ith artifact, $c_i \in \mathcal{C}$ for some context set $\mathcal{C}$. The contextual inspiration set is the subset that pertains to a particular context c, i.e. $\mathcal{I}_c = \{(z_i, c_i) :\ni c_i = c\}$. An artifact in the inspiration set could in general be associated with multiple contexts, and could involve a potentially complex interplay of various constituent components. Examples of inspiration sets of this type include recipe repositories tagged with cuisine information, a database of songs with their genres, etc. In the following section, we present an application of contextual computational creativity that highlights the use of generalization in the unsupervised setting.

Application: Creative Causal Associations

Analysts in domains such as financial, business, or intelligence analysis are often expected to use their creativity to imagine future worlds. Computational creativity methods could help analysts with divergent thinking, which is an important frame of mind for analyzing long-term and wide-ranging eventualities for scenario analysis (Heuer and Pherson 2010, p. 133). We describe an application in creative causal association that could spark ideas about future events. We explain the steps of our workflow as shown in Figure 1, where a dataset of political events is utilized for generating creative pairs of causally associated events in a country.

Causal Discovery: Building the Inspiration Set

Event Dataset. In relational (also known as dyadic) event datasets, events take the form ‘*who* does *what* to *whom*’, i.e. an event z involves a source actor a_z performing an action/verb v_z on a target actor a'_z, denoted $z = (a_z, v_z, a'_z)$.

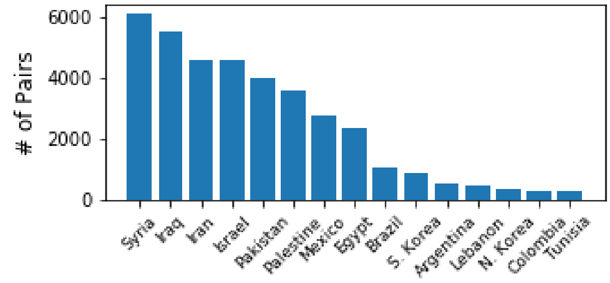


Figure 2: Bar chart showing the number of pairs (artifacts) in the inspiration set for 16 out of the 17 countries in scope. (India with over 45K pairs is omitted from this chart.)

The political science community has been building and curating such datasets for decades; see Schrodt and Yonamine (2013) for a review. While early datasets were obtained through human coding, this has been replaced by automated natural language processing methods that convert news articles in multiple languages into events.

For our application, we use the machine-generated Integrated Crisis Early Warning System (ICEWS) political event dataset (O'Brien 2010), with actors and actions from the Conflict and Mediation Event Observations (CAMEO) ontology (Gerner et al. 2002). Actors in this ontology could either be associated with generic actor roles and organizations (ex: *Police (Brazil)*) or they could be specific people (ex: *Hugo Chavez*). Actions in the CAMEO framework are hierarchically organized into 20 high-level actions and they can be classified by whether they are verbal or material and whether they involve cooperation or conflict.

In our experiments, we restrict attention to events that occurred in India and the 16 countries mentioned in Figure 2 in the time period 1/1/2011 – 12/31/2015. These are primarily countries from Asia and South America, and were chosen to try to find interesting interactions among actors within and across countries. The data was filtered to only include 17 actor roles, including *Citizen*, *Head of Government*, *Protester*, *Insurgent*, etc. Some manual curation was required to transform individuals who are current or former heads of government into their corresponding roles.

Causal Association. Causal discovery is a subject of great interest in AI and broadly across the sciences (Pearl 2009). Discovering causal association between a pair of events is typically done through human assessments (Singh et al. 2002) or learned from textual corpora (Radinsky, Davidovich, and Markovitch 2012; Luo et al. 2016). In this work, we have access to a structured *event* dataset of the form $\{(e_k, t_k)\}_{k=1}^N$, where e_k is the event type and t_k is the time of occurrence, $t_k \in R^+$. The dataset is strictly temporally ordered with initial time $t_0 = 0$ and end time $t_{N+1} = T$, where T is the total time period. We attempt to discover pairwise causal association by exploiting the fact that an event dataset can be modeled as a temporal point process and therefore represented using a conditional intensity model (Gunawardana and Meek 2016).

We make a simplifying modeling assumption: for a candidate cause-effect pair (x, y), suppose that the intensity of y at any time only depends on whether at least one event of type x has occurred in a preceding fixed window w. It can be shown that like the base rate of the effect λ_y, the conditional intensity parameter $\lambda_{y|x}^w$ can also be computed using summary statistics:

$$\lambda_y = \frac{N(y)}{T}; \lambda_{y|x}^w = \frac{N^w(x \leftarrow y)}{D^w(x)}, \tag{1}$$

where $N(y)$ counts occurrences of event y, $N^w(x \leftarrow y)$ counts occurrences where y occurs and at least one event of type x occurs within the preceding feasible time window w, and time period $D^w(x) = \sum_{k=1}^{N+1} \int_{t_{k-1}}^{t_k} I_x^w(t)dt$. Here $I_x^w(t)$ is an indicator for whether x has occurred at least once in a feasible window w preceding time t.

We propose a causal association score for the pair (x, y) that measures how the conditional intensity of effect y is modified by the presence of potential cause x. We refer to this score as the *conditional intensity ratio* with respect to the base rate, $CIR_B(x, y) = \lambda_{y|x}^w / \lambda_y$. We compute these scores for all event pairs for all 17 countries under consideration in one pass each through the country-specific datasets, using window $w = 15$ days and a minimum co-occurence $N^w(x \leftarrow y) = 20$ over the $T = 5$ year time period. We further filter out those pairs in a country whose scores are less than the mean score for that country. This process yields an inspiration set of causal pairs (x, y), counts of which are shown in Figure 2. India has the maximum number of events in ICEWS and ends up with at least one order of magnitude more pairs than any other country in scope.

Rule Generalization with Knowledge Graphs

There are many approaches to learning general relationships from artifacts in the inspiration set. Here we propose the use of knowledge graphs whenever available and relevant. Knowledge graphs, represented $\mathcal{G}(V, E)$, involve vertices V for entities (such as people, places, objects, etc.) and edges E that represent relationships between entities. Large-scale graphs such as DBPedia, Yago, Freebase, and the Google Knowledge Graph are popular in a host of applications; see Nickel et al. (2016) for a review.

Figure 3 provides a partial knowledge graph for our application, where the vertices include actors from Argentina and Brazil. Consider the following causal pair in the inspiration set for Brazil: *Govt (Brazil) Express Intent To Cooperate Govt (Argentina)* $\rightarrow$ *Citizen (Brazil) Disapprove Govt (Brazil)*. This event pair could potentially be generalized by finding paths in the knowledge graph from every actor in the event pair to the country of Brazil. The bold paths in the figure highlight two paths from *Govt (Argentina)* to *Brazil*, one from the neighbor relationship between Argentina and Brazil, and the other from the fact that they are both in the continent of South America. The resulting rule created from the former path is: *isGovtOf (country) Express Intent To Cooperate isGovtOf (isNeighborOf (country))* $\rightarrow$ *isCitizenOf*

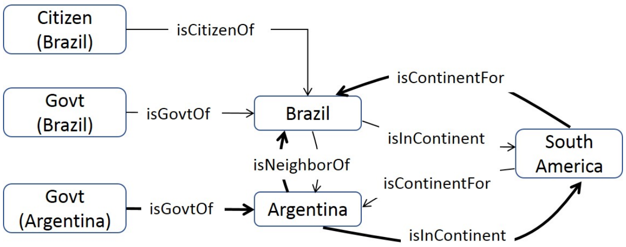


Figure 3: An example partial knowledge graph for selected actors and countries. The two cycle-free paths from *Govt (Argentina)* to *Brazil* are highlighted with bold arcs.

(country) Disapprove isGovtOf (country). Note that the instance from the inspiration set has been generalized and now potentially applies to any country. Similar abstraction paths on graphs pertaining to events have been referred to as predicate projections and have been used for prediction (Radinsky, Davidovich, and Markovitch 2012).

One could proceed in this fashion, compiling rules for all artifacts in the inspiration set $\mathcal{I}$ into a complete list of rules $\mathcal{R}$. We refer to the total number of times a rule r appears in $\mathcal{R}$ as its *support*, denoted $s(r)$.

For our implementation, we constructed an expanded version of the knowledge graph in Figure 3, reproducing similar relations for each of the 17 countries. Aside from the neighboring relation between bordering countries and membership in continents as well as sub-regions (Middle East and South Asia), we also included bi-lateral country relations of alliance (ex: Iran and Palestine) and enmity (ex: India and Pakistan) as they seem particularly suitable for CAMEO coded events of conflict and cooperation.

Causal Pair Generation

The final stage in our workflow is the generation of creative cause-effect pairs, in which we include the critical aspect of evaluating the quality and novelty of any arbitrary pair.

Evaluation. In this unsupervised setting, we estimate quality using the generalization rules. Specifically, if we denote the set of distinct generalized rules satisfied by event pair (x, y) as $\mathcal{R}_{xy} \subseteq \mathcal{R}$, then:

$$q(x, y) \propto \sum_{r \in \mathcal{R}_{xy}} s(r). \qquad (2)$$

Thus, our proxy for quality is the total support, which is a measure of how well a causal pair generalizes in aggregate across contexts. Note that according to the proposed metric, specific versions of rules are scored higher than their generalizations.

There are several ways of evaluating the novelty of an artifact in problems of contextual computational creativity. A reasonable approach is to compare the components of the artifact under consideration with those prevalent in the context. In our application, artifacts involve events with actors and actions; we consider an event contextually novel if the frequencies of the source actor, action, and target actor are low in the contextual inspiration set. The novelty of an event pair averages over both events in the pair. Specifically:

$$n_c(x, y) = \frac{g_c(a_x, v_x, a'_x)}{2} + \frac{g_c(a_y, v_y, a'_y)}{2}, \qquad (3)$$

$$g_c(a_z, v_z, a'_z) = (1 - f_c^s(a_z))(1 - f_c^v(v_z))(1 - f_c^t(a'_z)), \quad (4)$$

where $f_c^k(.)$ denotes the frequency of the component type k (either source actor s, action v or target actor t) in events in the inspiration set $\mathcal{I}_c$. The maximum novelty score is 1 and occurs when both events in a causal pair only include actors and actions that are not present in $\mathcal{I}_c$. Other approaches to measuring novelty are possible but not considered here.

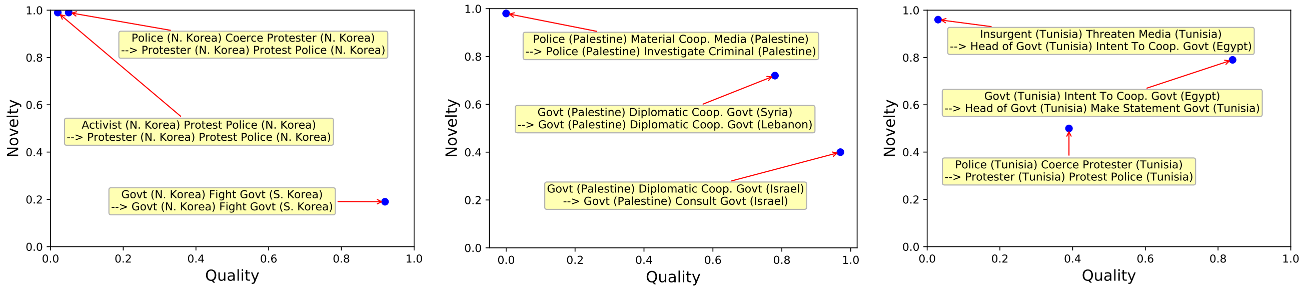
Generation Methodology. We generate creative causal pairs for a particular country (context) by first constructing a large set of instances from the complete list of rules $\mathcal{R}$. For every rule, we generate potentially many candidate pairs by traversing backwards on the knowledge graph $\mathcal{G}(V, E)$ from the country under consideration to identify actors along all relation paths in the rule. When a node has many parents that satisfy a particular relation, we randomly choose one of the parents as we walk on the graph. As an example, note that the relation path *isGovtOf (isNeighborOf (country))* from Iraq could lead to *Govt (Syria)* or *Govt (Iran)*. For our experiments, we generate up to 10 unique instances for every rule using these ‘random walks’, similar to Varshney, Wang, and Varshney (2016).

Once the candidate pairs have been generated, they can be exhaustively evaluated for quality and novelty and then aggregated/ranked in any desired fashion. We normalize quality scores by dividing by the maximum quality pair in a country; novelty is already normalized between 0 and 1.

Selected Results & Observations

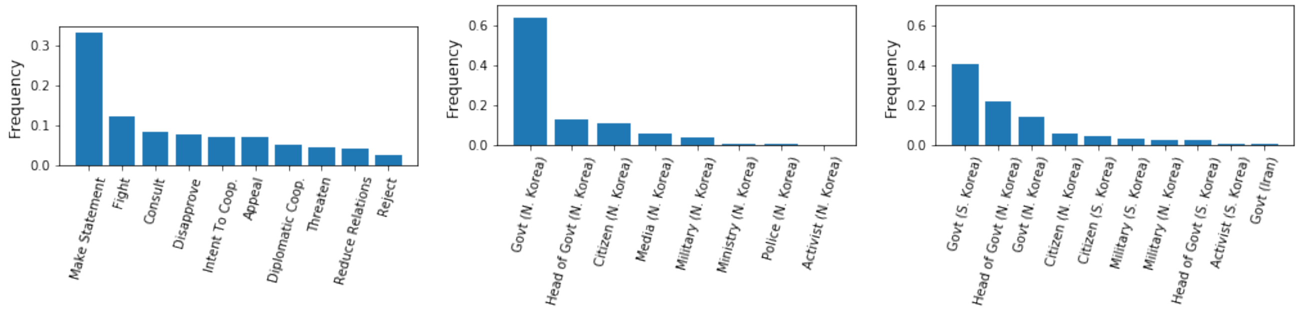
The ranked causal pairs could be used in a variety of ways. For instance, an analyst may wish to review high novelty pairs for assistance in conjuring up future possibilities in a country. Note that by construction, every pair satisfies at least one rule, so there is a minimum quality threshold applied to every pair.

Figure 4 shows selected pairs on a quality-novelty scatter plot from 3 countries in different regions of the world – North Korea, Palestine, and Tunisia. In North Korea (Figure 4 (a)), there is a pair around reinforcement of fighting that is deemed to be high quality as it generalizes well across countries. The two high novelty pairs are perhaps more interesting and involve protests; in one, these are brought on by activism while in the other, protests are caused by police coercion. Recall that we compute novelty using the frequencies of the artifact components in the inspiration set, which in our case are the actions, source actors, and target actors of events. The 10 most frequent components of each type for North Korea are shown in Figure 5. We observe that actions of protest and actors such as protesters and activists are rare in North Korea, which is why they are scored as novel by the system. An analyst may regard large-scale protests in North Korea to be implausible in the near future, yet engaging with



(a) North Korea (b) Palestine (c) Tunisia

Figure 4: Selected causal pairs on a quality-novelty scatter plot for 3 countries.



(a) Actions (b) Source actors (c) Target actors

Figure 5: Frequencies of the 10 most frequent actions, source actors, and target actors in the inspiration set for North Korea.

the system in this fashion could potentially generate useful ideas and directions for their investigation.

For the sake of comparison, we also plot the pair involving police coercion leading to protests for Tunisia (Figure 4 (c)). A quick investigation reveals that this pair is not as novel in Tunisia since protests are commonplace in the inspiration set. We see instead that pairs where the Tunisian government intends to cooperate with the Egyptian government are deemed novel. Note that Tunisia has the smallest inspiration set (Figure 2).

Due to the focus on country relations in our knowledge graph, we consistently observe high quality pairs involving actions such as diplomatic cooperation, consultation, and the making of public statements across countries; see for instance the high quality pairs in Palestine (Figure 4 (b)). Occasionally, the system is able to identify unusual and quirky pairs, such as the novel pair in Palestine where police cooperation with the media results in a criminal investigation.

We summarize a few other observations from our experimental investigation:

1. The knowledge graph could potentially result in numerous similar instances of the same rule being generated. In our case, group relations such as regional and continental membership result in pairs that appear repetitive as they differ only in the interacting foreign country. This effect could be limited by enforcing further filtering of pairs through additional restrictions and/or by deploying a *variety* module while recommending a set of pairs.

2. Another effect of the choice of knowledge graph, together with the choice of inspiration set, is that generated pairs for a context depend critically on the presence of existing relevant relations. For instance, our omission of USA for this analysis affects Mexico heavily – not having any associations with other countries in our knowledge graph, all of its recommended pairs only involve domestic actors.

3. Due to the aforementioned reasons, the current version of the system does require some human selection, much like other extant systems. The advantage of our proposed approach however is that the system is at least able to self-evaluate artifacts.

We highlight numerous challenges associated with our application. First, building a good inspiration set is difficult because the original data sources are machine-generated and noisy, not to mention the difficulty in discovering causal relations from statistical associations in an event dataset. Furthermore, acquiring and utilizing the appropriate knowledge is essential to the success of learning useful patterns/rules from the inspiration set. The current system is an early foray into work on creative scenarios; we believe that additional progress is required before the system's creations can be usefully evaluated by users.

Discussion

We discuss how the methods described in the previous section are more general than the application as well as the contextual computational creativity framework that was outlined. We also briefly make connections to a few other relevant concepts in computational creativity.

Generalizing Generalization (for Evaluating Quality). In the causal association application, we tried to identify and generate contextually creative artifacts by discovering artifacts that can be deemed novel for a particular context but also satisfy broader relationships learned from artifacts across contexts. Generalizing from the inspiration set could be used to evaluate quality for a broader class of computational creativity endeavors and could therefore be used in other types of applications.

Consider for example the application of creative recipes in culinary art. Varshney, Wang, and Varshney (2016) describe an approach that uses a knowledge graph pertaining to ingredients, which could include information about chemical compounds, seasonality, weather conditions pertaining to ingredient production, etc. One could use techniques similar to those described in the causal association application to generalize from such a knowledge graph along with an inspiration set of recipes, learning rules about which ingredients work well together based on edges (relations) in the graph. Varshney, Wang, and Varshney (2016) do indeed describe an association rule mining approach for learning patterns but they do not make the explicit connection to quality evaluation as we have done here. Association rule mining is one of several potential approaches for generalizing from artifacts that are represented as a set of constituent components, but note that artifacts could be modeled as more complex representations and that relations in such representations could also be generalized in numerous ways.

Contextual Creativity as P-Creativity. Boden (1990) distinguishes between *p (psychological)* and *h (historical)* creativity – the former refers to artifacts or ideas creative for a particular individual whereas the latter considers creativity from a historical perspective. In the contextual computational creativity framework outlined here, the intent is to be p-creative in a context by learning from history, through the inspiration set, perhaps along with other knowledge.

Generalization for Transformational Creativity. Boden (1990) also makes a distinction with regard to searching for artifacts, referring to producing combinations of familiar ideas and exploring the conceptual space as *combinatorial* and *exploratory* creativity respectively. She regards *transformational* creativity as transforming a conceptual space, such as by adding dimensions or changing constraints.

We highlight that using generalization to evaluate quality could potentially lead to behavior resembling transformational creativity in CC systems, at least in some ways. Injecting new data that is substantially different into the inspiration set could have the effect of modifying the way quality is evaluated and could therefore change constraints during search. Importantly, new knowledge acquired from data sources or other agents could have a more radical effect that alters the way in which quality is assessed.

A Note on Typicality. Ritchie (2001) mentions *typicality* of artifacts as another non-novelty related attribute that could be important in a CC system. We have ignored typicality in our application as it is partially built into the generation methodology, like in Morris et al. (2012) – actors that are associated with a particular country can be deemed typical for that context. It may however be useful to incorporate it more explicitly in our application, since one way to remove seemingly redundant cause-effect pairs is to screen out those that seem atypical by only considering a country's frequently associated foreign actors.

Conclusions

Evaluation is crucial in CC systems since an agent must be able to assess quality. In particular, the assessment function must work for previously unseen artifacts, since novelty is the whole point of creativity. In this paper, we have expounded upon the role that supervision plays in computational creativity by associating it with quality evaluation. Supervision could occur by directly encoding a quality function in a suitably abstract way, but it could also be learned through supervised learning algorithms.

We have proposed generalization as a means to evaluate quality in the unsupervised setting where quality is not specified in any explicit fashion. The benefits of unsupervised generalization in practical CC systems will likely primarily arise when used in conjunction with supervision from other agents. Furthermore, different generalization approaches may be suitable for different types of applications based on artifact and knowledge representations.

The core technical contribution of generalizing with a knowledge graph has been presented in a contextual computational creativity framework, where quality is determined from generalization that borrows strength from artifacts across contexts whereas novelty is context-specific. We imagine that this sort of approach may not be particularly useful when all contexts are similar in the inspiration set, since there would be little capacity to learn something new for any particular context.

We presented a detailed study of an application with cause-effect pairs of political events as artifacts and countries as contexts. Significant work remains towards graduating the proposed techniques in the workflow for the application into a full-fledged CC system. Suitable datasets and better models for causal discovery are essential, aside from improvements in the computational creativity techniques.

Acknowledgments

We thank our collaborators on the causal association research, Tian Gao and Nicholas Mattei, as well as three anonymous reviewers for their feedback. This work was supported in part by the IBM-Illinois Center for Cognitive Computing Systems Research (C3SR), a research collaboration as part of the IBM AI Horizons Network.

References

Bhattacharjya, D. 2016. Preference models for creative artifacts and systems. In *Proceedings of the 7th International Conference on Computational Creativity (ICCC)*, 52–59.

Boden, M. 1990. *The Creative Mind: Myths and Mechanisms*. London, UK: Weidenfield and Nicolson Ltd.

Buchanan, B., and Livingston, G. 2004. Toward automated discovery in the biological sciences. *AI Magazine* 25(1):69–84.

Colton, S., and Wiggins, G. A. 2012. Computational creativity: The final frontier. In *Proceedings of the 20th European Conference on Artificial Intelligence (ECAI)*, 21–26.

Colton, S. 2008. Automatic invention of fitness functions with application to scene generation. In *Proceedings of the EvoMusArt Workshop*.

Colton, S. 2012. The Painting Fool: Stories from building an automated painter. In McCormack, J., and D'Inverno, M., eds., *Computers and Creativity*. Berlin, Germany: Springer-Verlag. 3—38.

Cope, D. 1996. *Experiments in Musical Intelligence*. A-R Editions, Inc.

Diaz-Agudo, B.; Gervas, P.; and Gonzalez-Calero, P. A. 2002. Poetry generation in COLIBRI. In *Advances in Case-Based Reasoning: 6th European Conference on Case Based Reasoning*, 73–87.

Gerner, D. J.; Schrodt, P. A.; Yilmaz, O.; and Abu-Jabr, R. 2002. Conflict and mediation event observations (CAMEO): A new event data framework for the analysis of foreign policy interactions. *International Studies Association (ISA) Annual Convention*.

Gunawardana, A., and Meek, C. 2016. Universal models of multivariate temporal point processes. In *Proceedings of the Nineteenth International Conference on Artificial Intelligence and Statistics (AISTATS)*, 556–563.

Guzdial, M., and Riedl, M. O. 2018. Combinets: Learning new classifiers via recombination. arXiv:1802.03605.

Heuer, R. J., and Pherson, R. H. 2010. *Structured Analytic Techniques for Intelligence Analysis*. CQ Press.

Livingston, G. 2001. *A Framework for Autonomous Knowledge Discovery from Databases*. Ph.D. Dissertation, University of Pittsburgh.

Luo, Z.; Sha, Y.; Zhu, K. Q.; Hwang, S.-w.; and Wang, Z. 2016. Commonsense causal reasoning between short texts. In *Principles of Knowledge Representation and Reasoning (KR)*, 421–430.

Machado, P., and Cardoso, A. 2002. All the truth about NEvAr. *Applied Intelligence* 16(2):101–118.

Mayer, R. E. 1999. Fifty years of creativity research. In Sternberg, R. J., ed., *Handbook of Creativity*. Cambridge, UK: Cambridge University Press. 449–460.

McCorduck, P. 1990. *AARON's Code: Meta-Art, Artificial Intelligence and the Work of Harold Cohen*. New York, USA: Freeman.

Morris, R.; Burton, S.; Bodily, P.; and Ventura, D. 2012. Soup over bean of pure joy: Ruminations of an artificial chef. In *Proceedings of the 3rd International Conference on Computational Creativity (ICCC)*, 119–125.

Nickel, M.; Murphy, K.; Tresp, V.; and Gabrilovich, E. 2016. A review of relational machine learning for knowledge graphs. *Proceedings of the IEEE* 104(1):11–33.

Norton, D.; Heath, D.; and Ventura, D. 2010. Establishing appreciation in a creative system. In *Proceedings of the 1st International Conference on Computational Creativity (ICCC)*, 26–35.

O'Brien, S. P. 2010. Crisis early warning and decision support: Contemporary approaches and thoughts on future research. *International Studies Review* 12:87–104.

Pearl, J. 2009. *Causality: Models, Reasoning and Inference*. New York, NY, USA: Cambridge University Press, 2nd edition.

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*, 129–137.

Pinel, F.; Varshney, L. R.; and Bhattacharjya, D. 2015. A culinary computational creativity system. In *Computational Creativity Research: Towards Creative Machines*. Atlantis Thinking Machines, Springer. 327—346.

Radinsky, K.; Davidovich, S.; and Markovitch, S. 2012. Learning to predict from textual data. *Journal of Artificial Intelligence Research* 45:641–684.

Ritchie, G. 2001. Assessing creativity. In *Proceedings of the AISB Symposium on Artificial Intelligence and Creativity in Arts and Science*, 3–11.

Schrodt, P. A., and Yonamine, J. E. 2013. A guide to event data: Past, present, and future. *All Azimuth: A Journal of Foreign Policy and Peace* 2(2):5–22.

Singh, P.; Lin, T.; Mueller, E. T.; Lim, G.; Perkins, T.; and Zhu, W. L. 2002. Open mind common sense: Knowledge acquisition from the general public. In *On the Move to Meaningful Internet Systems 2002: CoopIS, DOA, and ODBASE)*, 1223–1237.

Toivonen, H., and Gross, O. 2015. Data mining and machine learning in computational creativity. *WIREs Data Mining Knowledge Discovery* 5:265–275.

Varshney, L. R.; Pinel, F.; Varshney, K. R.; Bhattacharjya, D.; Schorgendorfer, A.; and Chee, Y.-M. 2013. A big data approach to computational creativity. arXiv:1311.1213.

Varshney, L. R.; Wang, J.; and Varshney, K. R. 2016. Associative algorithms for computational creativity. *The Journal of Creative Behavior* 20(3):211–223.

Ventura, D. 2016. Mere generation: Essential barometer or dated concept. In *Proceedings of the 7th International Conference on Computational Creativity (ICCC)*, 17–24.

Wiggins, G. A. 2006. A preliminary framework for description, analysis and comparison of creative systems. *Knowledge-Based Systems* 19(7):449–458.

INES: A reconstruction of the Charade storytelling system using the Afanasyev Framework

Eugenio Concepción and **Pablo Gervás** and **Gonzalo Méndez**
Facultad de Informática
Instituto de Tecnología del Conocimiento
Universidad Complutense de Madrid
{econcepc,pgervas,gmendez}@ucm.es

Abstract

The present paper introduces INES (Interactive Narrative Emotional Storyteller), an instance of the Afanasyev story generation framework that rebuilds Charade, an agent-based storytelling system. The construction of INES pursues a double goal: to develop a more complete version of Charade, by including a plot generation stage; and to show the capability of Afanasyev as scaffolding for building united systems from sources of diverse kind. From a broad view, the resulting architecture is a microservice-oriented ecosystem in which every significant stage of the story generation process is implemented by a microservice that can be easily replaced by another, as long as the new microservice keeps the interface contract established by the Afanasyev model.

Introduction

Automatic story generation is a part of a wider research area in Artificial Intelligence named Computational Creativity (CC), which is the pursuit of creative behaviour in machines (Veale 2013).

A story generator algorithm (SGA) refers to a computational procedure resulting in an artefact that can be considered a story (Gervás 2012). The term story generation system can be considered as a synonym of storytelling systems, that is, a computational system designed to tell stories.

The operation of the story generation systems requires large amounts of knowledge. These systems are faced with a significant challenge of acquiring knowledge resources in the particular representation formats that they use. They meet an inherent difficulty when using formal languages in the detachment between the formulation of the needs in the real world and its representation in a formal construction. A possible solution can be the use of a Controlled Natural Language (CNL) for knowledge interchange (Concepción et al. 2016). This is precisely the approach introduced by the Afanasyev framework (Concepción, Gervás, and Méndez 2018). Afanasyev is a collaborative architectural model for automatic story generation which relates to a service-oriented architecture (Concepción, Gervás, and Méndez 2017a). It introduces an agnostic story representation model (Concepción, Gervás, and Méndez 2017b) that intends to ease the collaborative interchange of knowledge between different systems.

INES (Interactive Narrative Emotional Storyteller) is a reconstruction of Charade (Méndez, Gervás, and León 2016) based on the Afanasyev Framework. The original Charade system is a simulation-oriented agent-based story generation system. Charade was focused on generating stories about the evolution of the relationships between characters by running an unrestricted low-level simulation. The development of INES introduces a new stage in the Charade generation model, that is the plot generation. This stage provides the system with a more structured way of building the stories. Also, the development of INES allows for testing the suitability of the Afanasyev architectural structure and its knowledge representation model in a real-world context.

Background

The first story generation systems date back to the 1970s. The **Automatic Novel Writer** (Klein 1973) is considered to be the first storytelling system. It generated murder stories in a weekend party setting by means of generation grammars. **TALE-SPIN** (Meehan 1977) was another of the earlier story generators. It generated stories about the inhabitants of a forest. TALE-SPIN was a planning solver system that wrote up a story narrating the steps performed by the characters for achieving their goals. **Author** (Dehn 1981) was the first story generator to include the authors goals as a part of the story generation process. To this end, it intended to emulate the mind of a writer. From a technical point of view, Author also was a planner but, unlike TALE-SPIN, it used the planning to fulfill authorial goals instead of character goals. **Universe** (Lebowitz 1984) generated the scripts of a TV soap opera episodes in which a large cast of characters played out multiple, simultaneous, overlapping stories that never ended. In contrast with Author, Universe gave a special importance to the creation of characters, as it considered they were the driving force for generating stories. **Brutus** (Bringsjord and Ferrucci 1999) was a system that generated short stories using betrayal as leitmotiv. The main contribution of Brutus was its rich logical model for representing betrayal. This feature, along with its grammar-based generation component and its literary beautifier allowed it to generate quite complex stories. The **Virtual Storyteller** (Faas 2002; Swartjes 2006) is a Multi-Agent System that can generate stories by simulating a virtual world in which characters modeled by agents pursue their goals. In this way, the story emerges from the events in the virtual world. **Fabulist** (Riedl and Young 2010) is a complete architecture for automatic story generation and presentation. Fabulist combines an author-centric approach together with a representation of characters intentionality.

Although there is not much specific literature on the subject, there are some noticeable efforts concerning the reconstruction of an existing story generation that have been carried out. **Minstrel** (Turner 1993) was a story generation system that told stories about King Arthur and his Knights of the Round Table. Each story was focused on a moral, which also provided the seed for developing the story. Minstrel was developed in Lisp (Berkeley and Bobrow 1966) and used an extension of a Lisp library called Rhapsody (Malkewitz and Iurgel 2006) for representing the knowledge required by the generation process.

Skald (Tearse et al. 2014) is a publicly-released rational reconstruction of Minstrel for analysing original Turner's work in search of new implications for future research. Skald is written in Scala, a functional programming language that runs over the Java Virtual Machine. It is based on a previous project named **Minstrel remixed** (Tearse et al. 2012), that tried to develop a collection of improvements over the original Minstrel. The original components of Minstrel, as described in Turner's dissertation (Turner 1993), are the starting point for the Skald design. The work developed in Skald can be considered not only a collection of enhancements over Minstrel but a globally different picture of the original Minstrel, as well as a new system that sets the stage for future research in story generation.

One of the Skald key findings is an improvement over Minstrel's limitations: the story library, story templates, and the recall system must be tailored to one another for the original system to function. Tearse (2012; 2014) shows that this can be mitigated through a number of techniques, such as adding differential costs to transformations to remove the least-successful author-level actions.

Although Skald contains a good number of enhancements over Minstrel remixed, these are all aimed to expand its capabilities in a few areas: transparency in story generation, exploration and measurement of the subtle workings of individual modules, improved stability, and better story output in terms of speed, size, and coherence.

Skald keeps the original Minstrel specifications, in the sense that it simulates the actions of a human when producing stories. Skald puts its novelty at a lower level, using different levels of modules for simulating different problems. For example, it uses low level simulations of problem solving processes, while authorial goals are simulated using modules in a higher level.

Materials and methods

Charade

Charade (Méndez, Gervás, and León 2014; 2016) models the relationship between two characters using their mutual affinities, and applies it for generating stories.

This system is an agent-based architecture developed using JADE (Java Agent Development Framework) (Bellifemine, Poggi, and Rimassa 1999). It consists of two types of agents: a Director Agent, that sets up the execution environment and creates the characters; and the Character Agents, one for each character of the story, whose interactions generate the story.

The main objective of the system was implementing an affinity model as decoupled as possible from the story domain, and testing it independently from other factors such as the environment in which the action takes place or the personality traits and emotional state of the characters. Due to this independence, it can be easily used to generate different kinds of stories.

The generator is based on a simulation of the characters' interaction. During the simulation, the characters perform actions that result in a variation of their affinity levels. According to the affinity level, the characters can be a couple, friends, indifferent, and enemies. Generation is independent of the domain; although, since it focuses on affinities, it works best in domains where this affinity makes sense. The simulation is not directed, so that it can not be considered to constitute a plot or a story by itself. The input includes a complete parametrization of possible actions, categorized according the type of relationship allowed for the characters, the simulated characters, and their relationships measured in terms of affinity. The output consists of a list of actions proposed by characters, and the response of their counterparts, that can accept or reject the proposals, with the variation of affinity between the characters involved. Despite no text being generated, it would be easy to use a template for generating a textual description.

Afanasyev

Afanasyev (Concepción, Gervás, and Méndez 2018; 2017a; 2017b) is a framework specifically designed for building service-based automatic story generation systems. From an architectural point of view, it is basically a collection of microservices orchestrated by a high-level service. Each service exposes their capabilities as REST-based APIs (Fielding 2000) and it understands and generates JSON messages. Due to the fact that the inner logic of any microservice can come from a different storytelling system, its interface must be adapted to match the required contract so the microservice can operate under the conditions specified by the framework. This is the reason why Afanasyev includes the definition of the common REST interfaces provided by the services and leaves to every particular system the details of the implementation.

The main microservices in Afanasyev, depicted in Figure 1, are the following:

- Story Director, the microservice that orchestrates the whole ecosystem.
- Plot Generator, the microservice that generates a high-level plot.
- Episode Generator, the microservice that fills the scenes that composed the plot.
- Filter Manager, the microservice that manages a set of filters that will be applied to the story each time it changes (due to the activity of the Episode Generator).

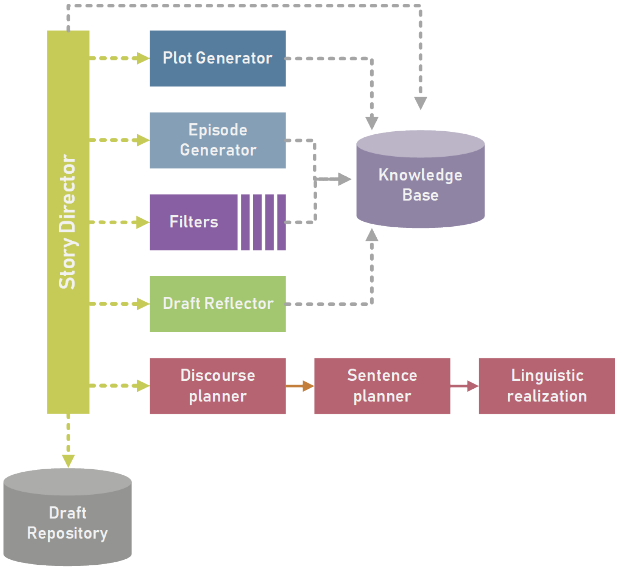


Figure 1: Architecture of Afanasyev.

- Draft Reflector, the microservice that analyses the story for deciding whether it is completed or not.
- Discourse generation services (Discourse Planner, Sentence Planner and Linguistic Realization), which turn the abstract story model into a human-readable text in Natural Language.

In order to allow the combined operation, the microservices of the framework require a common representation model for stories. The Afanasyev representation model (Concepción, Gervás, and Méndez 2017b) focuses on the knowledge that is directly related to the story, instead of that related to the generation process, which would be hard to export between different systems. The model has been designed as a hierarchical structure, in which the root concept is the **story**. Most of the leafs of this tree-like structure are assertions representing a piece of knowledge. These assertions are expressed by means of sentences in a Controlled Natural Language (CNL) (Schwitter 2010). In Afanasyev, every story is composed by a plot and a space. The plot represents the sequence of events —actions and happenings, that constitutes the skeleton of the story. The space encompasses the whole universe in which the story takes place, including the existents —characters, living beings and objects that take part in the story, and the setting —the set of locations mentioned in the story.

Persistence in Afanasyev is mainly composed by two stores: the Draft Repository and the Knowledge Base. The Draft Repository is a database that stores the ongoing drafts. The current implementation of this component is based on a NoSQL database (Han et al. 2011), namely MongoDB (2017). The knowledge base has the task of preserving all the knowledge related to concepts, relationships between concepts, rules, etc. It is a knowledge base generated from the contributions of the involved story generation systems. This model of knowledge syndication allows to increase the

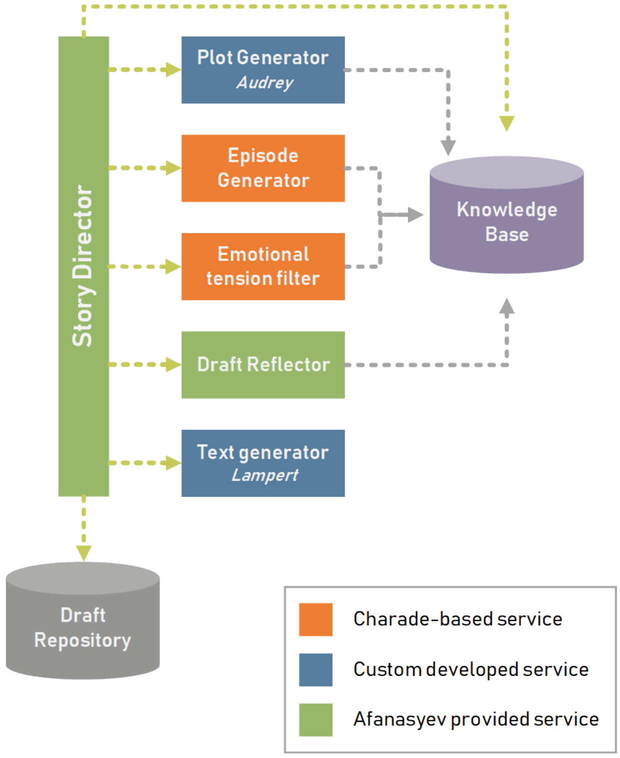


Figure 2: Architecture of INES.

shared set of concepts each time a new system joins the ecosystem. Hence, every contributor performs an initial load expressing its rules.

INES

INES is the translation of the Charade storytelling system to the Afanasyev architectural framework. The purpose of this work is two-fold: to validate the capability of the Afanasyev model for supporting different story generation models and to prepare the integration of Charade in a wider service-based collaboration ecosystem.

The main adaptation work has focused on a central aspect of the original Charade behaviour: the directed simulation. In effect, Charade originally produced outputs that were the result of an unrestricted simulation. In the case of INES, there is a preexisting plot to which the output of every simulation must be adapted. This means that, for each scene, there is a specification based on precondition / postcondition that implies that not every possible result of the simulation is valid.

The architecture of INES, as adapted from Afanasyev, is depicted in Figure 2. It is a combination of Afanasyev ready-made services, along with the specific Charade adapted services and a set of newly created services, required by the framework:

- Story Director, provided by Afanasyev
- Plot Generator, required by Afanasyev and newly developed for INES

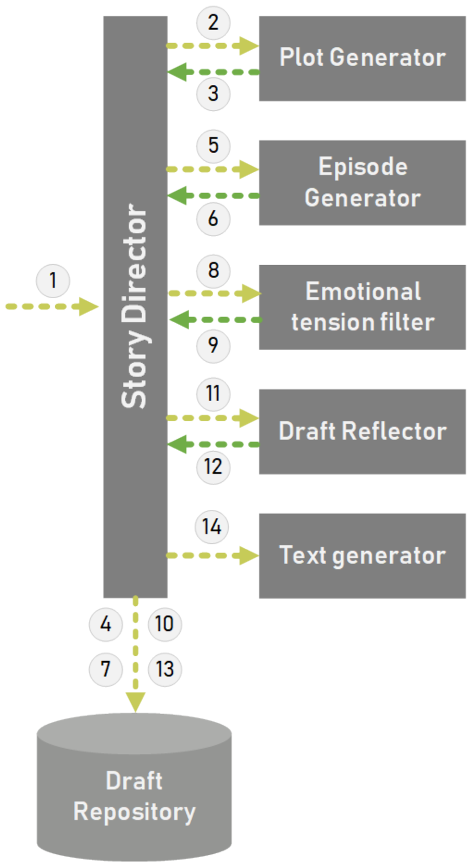


Figure 3: Operation of INES / Afanasyev.

- Episode Generator, created from the Charade system
- Emotional tension filter, created from the Charade system
- Draft Reflector, provided by Afanasyev
- Text generator, required by Afanasyev and newly created for INES

The Story Director

The architecture of Afanasyev is an ecosystem of microservices. The Story Director manages the joint operation of the whole ecosystem, as depicted in Figure 3. It orchestrates the execution of the different story generation stages by requesting the APIs of the different services. This processing proceeds iteratively, generating drafts that will be refined in each pass, until the established criteria for story completeness are met.

The first step consists in generating the basic structure of the plot. It is performed by the Plot Generator, that establishes the sequence of episodes that make up the plot. Each episode is interwoven with the others by means of its pre and post-conditions. These are collections of statements relating to the setting and the existents of the story.

The Plot Generator: Audrey

The **Plot Generator**, named **"Audrey"** —after Audrey Hepburn who played the lead role in "Charade", has been developed specifically for INES and is a template-based plot generator which produces outlines from a subset of the cinematographic basic plots compiled by Balló (Balló and Pérez 2007). Its basic procedure can be considered akin to those applied by systems like Gester (Pemberton 1989) and Teatrix (Machado, Paiva, and Brna 2001). The basic idea behind Audrey is building a story plot containing the main scenes that will be developed by the Charade-based Episode Generator. The plot building procedure starts by selecting one of the predefined templates, which consist of a conceptual structure with the shape of the plot. The template can be selected randomly or it can be picked according to the template name received as a parameter. Once a basic template is selected, Audrey gives it substance by instantiating the generic elements of such template. For achieving this, it requires to know about the context in which the story will be set. In this case, the context is inferred from the preconditions passed as parameters. These preconditions are a collection of assertions involving concepts that are necessarily kept in the knowledge base.

An example of one of these templates is "The destructive outsider". This story is essentially composed by the following episodes:

- The initial state: a peaceful community.
- The arrival of the outsider.
- The outsider acts against the members of the community, performing destructive actions, without being uncovered.
- The true evil nature of the outsider is revealed.
- The heroes rise from the community and fight against the outsider.
- The outsider is purged. The community becomes peaceful again.

In order to develop a consistent detail for every episode, Audrey requires a knowledge base that contains the main concepts presented in the plot. In this case, the plot mentions a "community", an "outsider", some "destructive actions" performed by the outsider, a group of "heroes" that rise against the outsider, and certain "purging actions" that the heroes perform. All these concepts are related to each other and can be represented by means of a graph. So, the required knowledge for instantiating the example is partially depicted in Figure 4.

So, the relationships between the concepts can be considered as assertions such as: "When the community is a family, then the outsider can be a new partner, an unknown relative and a new lover. When the outsider is a new partner, then the arrival can be a marriage".

The translation of these relationships to a physical database fits better with a graph-oriented database. In this particular case, the knowledge base has been implemented using Neo4j (Vukotic et al. 2014). So there are nodes with labels such as "Community", "Outsider", "Arrival" and "Action" representing the main plot concepts. The relationships

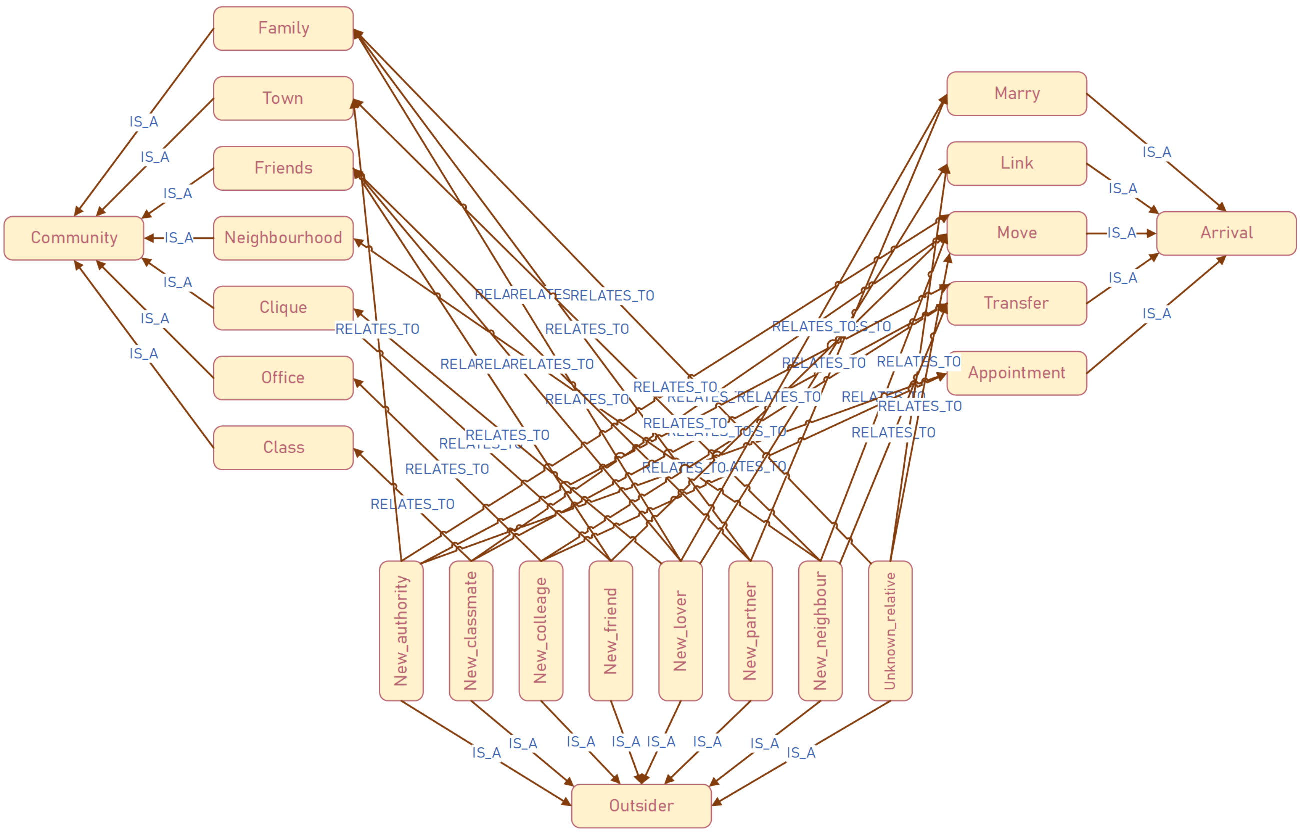


Figure 4: Partial view of the concepts relationships in the KB.

between the instances of the concepts are represented by means of the graph edges (i.e. database connections between nodes).

During the plot generation procedure, Audrey queries the knowledge base for extracting the possible instances for the concepts involved in the plot template. In this way, the concept “Outsider” is replaced by a “new neighbour” or a “new sheriff”, according to the setting in which the story takes place.

The next step concerning this knowledge is to apply it for determining the actions that the characters can perform during the simulation in order to keep the story consistency. For example, the knowledge base can label the acts of “insult” and “kill” as “hostile actions”. If the outsider has harmed the community by sowing discord, it would be unjustifiably excessive that the heroes reacted by killing him. In this case, the context of the story provides the episode generator with the appropriate actions that the actors could perform.

The Episode Generator

This microservice is based on the original Charade core. It generates a complete simulation of characters interaction according to the restrictions that are provided as input parameters. As mentioned in the previous section, the types of actions that can be consistently performed by the characters are limited by the context of the story. So, the episode generator receives not only the ongoing story, but also the preconditions and postconditions that the resulting simulation must match. This approach introduces a shift in the prior behaviour of the Charade’s engine, which originally drove an unrestricted simulation.

Charade was designed for obtaining the list of possible actions from its configuration, during system startup. It distinguished between three types of actions (love, friendship and enmity). In order to ease the adaptation of Charade as a microservice in the Afanasyev ecosystem, the set of possible actions are passed to it as part of the request parameters. For selecting the most suitable actions, the Story Director queries the knowledge base and retrieves the context-related actions that better fit the storyline. For example, continuing the previous example, if the plot is related to a family and the outsider has stolen something, the actions carried out in response to this offence could be “insult”, “report the burglary”, “demand the restitution” and “demand to leave”. These options would be retrieved and passed in the request as the proper actions that could be performed by the “heroes” of the story. Then, the episode generator would select some of them during the simulation and complete the detail of the episode. The current version of the episode generator preserves partially the randomness of the original Charade simulation model. In particular, it chooses randomly the actions performed by the characters from the set of allowed actions.

Table 1 shows a sample story that can be generated by applying this model.

Episode	Actions
A peaceful community	John invites William to dinner John invites Mary to dinner William helps John to cook Mary gives a present to John
The arrival of the outsider	John makes a welcome party for David (the outsider) David gives a present to John William helps David to move Mary helps David to move
Outsider destructive actions	David steals a valuable object from John's house David tells Mary that William is the thief
Conflict	Mary believes David Mary insults William William gets angry with Mary
The outsider revealed	William discovers David stealing in John's house William tells John that David is the thief John tells Mary that David is the thief
The rise of the heroes	John insults David John demands David to leave David leaves the town
Conclusion	Mary says sorry to William William gives thanks to John Mary gives thanks to John

Table 1: A sample story based on "The destructive outsider"

The Emotional tension filter

The current version of the Emotional tension filter works in a very simple way. It is a filter which is invoked after every episode simulation —performed by the Episode Generator, and it determines if the generated actions fit certain drama parameters. To meet this purpose, the Emotional tension filter considers the semantic information associated to the actions in the knowledge base to adjust the strength of the drama in the story. For example, considering again the story of "The destructive outsider" plot, an action such as "to slap" the outsider is much more dramatic than "to demand him to leave". By establishing the threshold for the tension, this service helps the Story Director to select the most dramatic continuation of the plot. So, this filter removes a subset of the generated episodes, and makes the Story Director to call again the Episode Generator until the whole plot has been adequately completed.

All the actions referenced in the knowledge base have a numerical attribute which reflects its intensity in terms of drama. This is a feature closely related to the original Charade operation (Méndez, Gervás, and León 2016; 2014). The higher the intensity of the action is, the higher is the numerical value. This representation helps the filter to decide whether an episode deserves to be included in the draft or not.

The Draft Reflector

The Draft Reflector of INES is the original basic Afanasyev-provided Draft Reflector. This microservice simply checks if all the episodes have been developed according to the plot restrictions. In this case, the choice is based on the need of keeping the draft analysis stage as simple as possible. The interest of the INES model is related to the ability of Afanasyev to provide a suitable architecture for building a system like Charade by means of its building blocks.

The Text Generator: Lampert

The other INES-specific service developed is the Text Generator, named "Lampert" —after Audrey Hepburn character's surname in "Charade". Lampert is microservice that translates the plot, represented as a data structure, into a text in Natural Language. Its core is based in the SimpleNLG Java library (Gatt and Reiter 2009).

The text generation is the last stage in the story generation process. Lampert has been designed simply for conveying the story represented in the Afanasyev common representation. Its purpose is not so much being a literary beautifier but providing a human-readable summary of the story.

Discussion

Afanasyev is not focused towards the ad hoc integration of specific pre-existing systems, but rather to provide a general service-oriented framework that allows the construction of different storytelling systems by assembling components from various systems (or from only one, in the simplest case). For this reason, the adaptation of Charade has required a number of transformations. Firstly, Charade has been designed as a lightweight agent-based architecture. Its essential logic has been preserved in INES, but the system operation has been restructured. The operation of every microservice in INES is completely independent from the others. Adapting the simulation flow has involved the development of a couple of new components that did not exist in Charade: the Plot Generator and the Text Generator. These two microservices could be easily replaced by other microservices based on different approaches that the current ones, as this is the essence of the Afanasyev framework.

The generation model of the Plot Generator is quite simple, but also very convenient for filling the existing gaps in the original model. It can be easily extended by providing more plot structure templates. Also, the richer the knowledge base is, the more interesting the generated stories are. As it has been shown, the role of the knowledge base is essential in this model for achieving coherent and believable stories. The same basic plot template can be instantiated in a wide spectrum of stories. As new instances are added to the database, the variability will increase accordingly.

Another relevant addition to the original Charade behaviour is the Emotional Tension Filter. It allows the system to generate stories with a greater drama, or not, depending on the filtering values. This service can be enhanced, or even replaced by a much more complex one, in order to help to create stories according to certain narrative tension curves. This configuration will entail a more global way of operation, considering not only the particular tension of an episode, but the evolution of the whole narrative arc.

Despite its simplicity, the Text Generator service provides a useful output. It has been deliberately designed for providing a summary in Natural Language rather than an elaborate literary text. Naturally, it can also be replaced by a much more complex surface realizer which provides a more polished literary work. A future candidate could be the TAP SurReal Surface Realizer (Hervás and Gervás 2009).

Conclusions and future work

The Afanasyev framework, despite having been originally conceived as an architectural model for building collaborative storytelling architectures, should not be seen solely as a tool for system integration. The purpose of developing INES was to prove that the Afanasyev framework can also be used for rebuilding any system as a microservice-based model. In this particular case, the adaptation of a purely agent-based simulation-oriented story generation system to a microservice-based pre-existing framework was particularly challenging. The resulting system can be considered an evolved version of the original Charade system, with a more structured approach to story generation.

Another interesting derivative of the work carried out during the design and development of INES is the knowledge base itself. It has been addressed using a representation model based on a graph-oriented database. This has allowed for simplifying the representation, as well as to use a general industry-oriented development stack, instead of a stack specifically oriented to Artificial Intelligence. The fact that the database can also be consulted by means of a REST interface provides an additional decoupling mechanism that will allow to evolve it independently, and even its replacement, without affecting the operation of the rest of the microservices ecosystem.

In the present version of INES, for every draft processed in every iteration, several continuations can be generated and added to the population of drafts to process during the next iteration. On the generated population, a reflection process is applied by means of the Draft Reflector microservice, and the drafts that it considers already finished are marked as stories. This process continues until all drafts are marked as finished or a limit of iterations is reached (to guarantee completion). In the face of future work, the development of a service that helps to decide what is the most appropriate level of detail in each of the scenes is still pending. This aspect can be provided in a first instance by a human —applying a co-creation model—, but it would be perfectly evolved to introduce a component for automating this task.

Acknowledgments

This paper has been partially funded by the projects IDiLyCo: Digital Inclusion, Language and Communication, Grant. No. TIN2015-66655-R (MINECO/FEDER) and InVITAR-IA: Infraestructuras para la Visibilización, Integración y Transferencia de Aplicaciones y Resultados de Inteligencia Artificial, UCM Grant. No. FEI-EU-17-23.

References

Balló, J., and Pérez, X. 2007. *La semilla inmortal: los argumentos universales en el cine*. Ed. Anagrama.

Bellifemine, F.; Poggi, A.; and Rimassa, G. 1999. Jade–a fipa-compliant agent framework. In *Proceedings of PAAM*, volume 99, 33. London.

Berkeley, E. C., and Bobrow, D. G. 1966. *The programming language LISP: Its operation and applications*. MIT Press.

Bringsjord, S., and Ferrucci, D. 1999. *Artificial intelligence and literary creativity: Inside the mind of brutus, a storytelling machine*. Psychology Press.

Concepción, E.; Gervás, P.; Méndez, G.; and León, C. 2016. Using cnl for knowledge elicitation and exchange across story generation systems. In *International Workshop on Controlled Natural Language*, 81–91. Springer.

Concepción, E.; Gervás, P.; and Méndez, G. 2017a. An api-based approach to co-creation in automatic storytelling. In *6th International Workshop on Computational Creativity, Concept Invention, and General Intelligence. C3GI 2017*.

Concepción, E.; Gervás, P.; and Méndez, G. 2017b. A common model for representing stories in automatic storytelling. In *6th International Workshop on Computational Creativity, Concept Invention, and General Intelligence. C3GI 2017*.

Concepción, E.; Gervás, P.; and Méndez, G. 2018. Afanasyev: A collaborative architectural model for automatic story generation. In *5th AISB Symposium on Computational Creativity. AISB 2018*.

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

Faas, S. 2002. Virtual storyteller: an approach to computational storytelling. *Unpublished masters thesis, University of Twente, Department of Electrical Engineering, Mathematics and Computer Science*.

Fielding, R. T. 2000. *Architectural styles and the design of network-based software architectures*. Ph.D. Dissertation, University of California, Irvine.

Gatt, A., and Reiter, E. 2009. Simplenlg: A realisation engine for practical applications. In *Proceedings of the 12th European Workshop on Natural Language Generation*, 90–93. Association for Computational Linguistics.

Gervás, P. 2012. Story generator algorithms. In *The Living Handbook of Narratology*. Hamburg University Press.

Han, J.; Haihong, E.; Le, G.; and Du, J. 2011. Survey on nosql database. In *Pervasive computing and applications (ICPCA), 2011 6th international conference on*, 363–366. IEEE.

Hervás, R., and Gervás, P. 2009. Evolutionary and case-based approaches to reg: Nil-ucm-evotap, nil-ucm-valuescbr and nil-ucm-evocbr. In *Proceedings of the 12th European Workshop on Natural Language Generation*, 187–188. Association for Computational Linguistics.

Klein, S. 1973. Automatic novel writer: A status report. *Papers in text analysis and text description*.

Lebowitz, M. 1984. Creating characters in a story-telling universe. *Poetics* 13(3):171–194.

Machado, I.; Paiva, A.; and Brna, P. 2001. Real characters in virtual stories. In *International Conference on Virtual Storytelling*, 127–134. Springer.

Malkewitz, S. G. R., and Iurgel, I. 2006. Technologies for interactive digital storytelling and entertainment. In *TIDSE*. Springer.

Meehan, J. R. 1977. Tale-spin, an interactive program that writes stories. In *In Proceedings of the Fifth International Joint Conference on Artificial Intelligence*, 91–98.

Méndez, G.; Gervás, P.; and León, C. 2014. A model of character affinity for agent-based story generation. In *9th International Conference on Knowledge, Information and Creativity Support Systems, Limassol, Cyprus*, volume 11, 2014.

Méndez, G.; Gervás, P.; and León, C. 2016. On the use of character affinities for story plot generation. In *Knowledge, Information and Creativity Support Systems*. Springer. 211–225.

2017. Mongodb official site. `https://www.mongodb.com/`. [Online; accessed 29-December-2017].

Pemberton, L. 1989. A modular approach to story generation. In *Proceedings of the fourth conference on European chapter of the Association for Computational Linguistics*, 217–224. Association for Computational Linguistics.

Riedl, M. O., and Young, R. M. 2010. Narrative planning: balancing plot and character. *Journal of Artificial Intelligence Research* 39(1):217–268.

Schwitter, R. 2010. Controlled natural languages for knowledge representation. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, COLING '10, 1113–1121. Stroudsburg, PA, USA: Association for Computational Linguistics.

Swartjes, I. 2006. The plot thickens: bringing structure and meaning into automated story generation.

Tearse, B. R.; Mawhorter, P. A.; Mateas, M.; and Wardrip-Fruin, N. 2012. Lessons learned from a rational reconstruction of minstrel. In *AAAI*.

Tearse, B.; Mawhorter, P.; Mateas, M.; and Wardrip-Fruin, N. 2014. Skald: minstrel reconstructed. *IEEE Transactions on Computational Intelligence and AI in Games* 6(2):156–165.

Turner, S. R. 1993. *Minstrel: A Computer Model of Creativity and Storytelling*. Ph.D. Dissertation, University of California at Los Angeles, Los Angeles, CA, USA. UMI Order no. GAX93-19933.

Veale, T. 2013. Creativity as a web service: A vision of human and computer creativity in the web era. In *AAAI Spring Symposium: Creativity and (Early) Cognitive Development*.

Vukotic, A.; Watt, N.; Abedrabbo, T.; Fox, D.; and Partner, J. 2014. *Neo4j in action*. Manning Publications Co.

Interview with the Robot: Question-Guided Collaboration in a Storytelling System

Philipp Wicke
School of Computer Science
University College Dublin
Dublin D4, Ireland
philipp.wicke@ucdconnect.ie

Tony Veale
School of Computer Science
University College Dublin
Dublin D4, Ireland
tony.veale@ucd.ie

Abstract

An automated storytelling system that presents its stories as text on a screen is limited in its engagement with readers. An embodied robotic agent can overcome these limitations by using gesture to physically enact its stories. This paper presents a robotic agent that enacts its own novel stories, which it shapes from the feedback it obtains using probing personal questions. Our robotic writer/presenter has two alternate modes of story-generation: a straight "telling" mode and an interview-oriented back-and-forth that extracts personal experiences from the user as raw material for new stories. We explore the practical issues of implementing both modes on a NAO humanoid robot that integrates gestural capabilities in an existing story-telling system.

Introduction

An intriguing story that captivates listeners is a product of multiple intertwined factors, from the complexity of the plot to the way the storyteller uses speech and gesture to interact with an audience. Ultimately, however, it is the emotional nature of the human listener, such as the listener's willingness to get emotionally involved in a story and to empathize with its characters, that leads to true appreciation of a narrative. This empathy can be triggered by identification (Krebs 1975) with the characters in the story or with the perspective conveyed by the storyteller. This path from identification to empathy to engagement requires that the listener be human, but it does not require the teller to be human too. This paper provides a system description of a creative story-generator that augments its symbolic narratives with embodied gestures using a robot for knowledge elicitation and story presentation. It explores two alternate modes of engagement that vary in the amount of personal listener experience that is integrated into a tale to foster identification and empathy.

As we use the Nao humanoid robot in this work (Fig. 1), we will briefly comment on the technical details of this physical platform, before surveying previous creative work with the Nao. Storytelling with a robot, as presented here, will combine story-generation with story-enactment. The former generates stories that make maximal use of the robot's affordances, while the latter executes these affordances to maximize user engagement. Enactment combines not just speech (textual delivery) but gesture, posture and body orientation. In short, robotic agents provide more dimensions of expressiveness than text or speech alone.

The following sections focus on a discussion of the technical difficulties that we have encountered, and on the techniques and resources we have used to overcome them. To investigate our hypothesis that a robotic platform can enhance listener identification with the products of automated story-generation, we contrast the implementation of two modes of user-influenced story-generation. The first, a baseline, employs simple user interaction prior the generation of a story, while the other conducts a probing, therapist-like interview to elicit personal experiences from the listener that it can repackage – somewhat collaboratively – into novel stories.

An empirical evaluation of the approaches employed is still in progress, so for now we conclude with insights from the current implementation to explain why we identify more with an embodied robot than with an intangible piece of software, and perhaps why are more likely to attribute creativity to the former than to the latter.

The Humanoid Nao

The French robotics company Aldebaran Robotics (recently acquired by Softbank) began to design its Nao humanoid robot (Fig.1) in 2004. Four years later the first release was provided for research use in universities and laboratories. The current model has a height of 58 cm, weighs 4.3 Kg and is powered by a 48.6 Wh lithium battery that provides approx. 90 minutes of active use. The bipedal robot has 25 degrees of freedom controlled by a 1.6 GHz CPU. The inbuilt Linux-based *NAOqi* OS gives access to two HD cameras, four microphones and a variety of sensors for detecting inertia, pressure and infrared light (Gouaillier et al. 2009). A detailed investigation of the robot's technical capabilities can be found in (Shamsuddin et al. 2011). With its wide variety of easily accessible tools, the robot quickly found its way into research (Tapus et al. 2012), (Parde et al. 2015) and education (Hood, Lemaignan, and Dillenbourg 2015). The robot provides in a ready-made set of modalities: Speech production and speech recognition software, facial recognition that allows the robot to follow faces with its gaze, a pre-installed set of 400+ body movements and although the robot has no facial gesture capabilities it has a set of RGB LEDs that function as eyes, which can show different colors to communicate emotions. The proceeding section will

Figure 1: The Nao robot from Aldebaran. In this image, the robot performs a gesture titled *ShowSky_1* that has been associated (with medium strength) to the actions: look up, praise, pray, preach and worship.

comment on studies investigating those modalities. The use of gestures is the key modality in the storytelling framework presented in this paper and we will therefore outline the importance of this modality through previous research.

The Nao's uses are myriad, so this section will highlight the most notable research that has been published in the domain of storytelling. An overview of storytelling with robots can be found in (Chen, Nurkhamid, and Wang 2011). For instance, (Ham et al. 2011) investigated the impact of deliberate gesture and gaze during storytelling by a Nao robot. They employed a database of gestures and gazing modes that were sourced from a professional stage actor and their results showed that the effect of using gesture and gaze together was significantly greater than the effect of using a single modality in isolation. The Nao has a gazing mode that is active by default, which causes it to autonomously shift its focus from one person to another, often to the person who happens to be speaking at any given moment. We can assume, based on (Ham et al. 2011), that this default gaze-shifting functionality enhances the communicative effect of its speech mode, regardless of whether other gestures are also used. Those authors relied on a small dataset of just 21 gestures, whereas the framework we present here employs more than 400 gestures, each of which has been associated with one or more story verbs.

The current framework emphasizes not just gesture and gaze but the use of language to elicit personal experiences from a user. We draw on studies from (Csapo et al. 2012), (Meena, Jokinen, and Wilcock 2012) and (Wilcock and Jokinen 2013) that use the Nao in question-answering mode. The framework described in that work marries Nao's dialogue capabilities with a question-answering interface for Wikipedia that allows a user to retrieve information from the web in a conversational manner (in much the same vein as Amazon's Alexa/Echo). The evaluation of human/robot interaction in (Csapo et al. 2012) focuses on modalities such as tactile sensors, face detection and non-verbal cues to derive these findings: the optimal communication distance is 0.9 meters between human and robot, while the limitations of Nao's speech recognizer make sudden interruptions inadvisable and highly impractical to process. The latter insight influences the constraints we describe in a later section on the modes of our own framework. Findings regarding the use of gesture are less relevant here, since those authors relied on a database of just six gestures for their system.

Gesture and gaze enrich the communication process, but how much accuracy do these non-verbal additions provide? (Häring, Bee, and André 2011) suggests that the Nao robot's body movements are most effective for communicating a specific set of emotions, whilst eye color and sound are evocative but much less accurate in conveying specifics. This leads us to prioritize physical gestures in our approach, while relying on colour and sound as non-vital embellishments. As each of the studies mentioned here suggests that gestures achieve a heightened effect when paired with the relevant speech content, we do not need to focus our evaluation on behavioral cues, but on the communicative aspects of the interaction with the user.

A comparable number of gestures to that employed here can be found in the work of (Pelachaud et al. 2010), (Le, Hanoune, and Pelachaud 2011) and (Gelin et al. 2010). Those authors developed two markup languages for use in functional and behavioral annotation for story-telling with a virtual animated avatar. They subsequently selected a subset of their database of approximately 500 annotated gestures for use in embodied storytelling with a Nao robot that can read stories to children. The empirical evaluation of their approach indicates that while the gestures are reported as appropriate to the content they adorn, they are not often seen as natural adornments to the action (Le and Pelachaud 2012).

Most approaches to automated story-generation lend themselves to robotic embodiment, insofar as any stream of action-oriented text can be augmented at suitable junctures with appropriate gestures and gaze behaviours. Thus, the engagement-reflection approach of (Pérez y Pérez and Sharples 2001), as implemented in the Mexica system, is as suited to the enactive mode of story presentation as the morphological approach (in the sense of Vladimir Propp's *morphology of the folk tale*) of (Gervás 2013) to the plot-as-planning-and-problem-solving approach of (Riedl and Young 2010). So we are principally guided by practicality rather than theory in our choice of the *Scéalextric* model of (Veale 2017) as the story-generating core of our system. *Scéalextric* employs an open and modular knowledge representation that is easily extended, and provides a public distribution that contains tens of thousands of story-related semantic triples to support plot and character design. It is built around an inventory of over 800 action verbs that gives our robot the semantic material to pose highly specific questions of a listener in its path to building a vivid story.

Some fascinating recent work in Computational Creativity has focused on humans and artificial agents working in unison to achieve *co-creativity* (Jordanous 2017). For example, (Davis et al. 2016) explored co-creativity in the domain of abstract drawing, using an enactment framework to identify the emergence of *sense-making* in a contrastive study of human-human *and* human-machine collaboration. A co-creative approach to storytelling is found in the *Mable* system of (Singh, Ackerman, and Pérez y Pérez 2017), which builds on the *Mexica* story generation system to write lyrics for a ballad that tells a tale. This system also builds on the *Alysia* system of (Ackerman and Loker 2017), which uses machine-learning to support the creation of melodies. The combined system first composes lyrics in a co-creative mode with a human user, and subsequently overlays these lyrics onto a machine-generated melody. Good music is designed to move the emotions *and* the body, so musical story-telling presents significant opportunities for physical embodiment in a robot. So the approach described here is one of many potential story-generation services (in the sense of (Veale 2013)) that can be selected from a competitive API economy (Concepción, Gervás, and Méndez 2017) for automated storytelling on demand. Another API service in this economy, the *Charade* system of (Méndez, Gervás, and León 2016), suggests obvious parallels to our current framework, insofar as it motivates the development of inter-character affinities. Each service will advertize its own comparative advantage, and so the current approach offers nuance in its use of enactment to unify vivid actions with embodied gestures.

The body is the means by which we engage with our physical environment, so an embodied story-teller whose gestures appear natural can grant a greater sense of reality to the wholly invented realms of its imaginary stories. Each of the studies considered here thus emphasizes the importance of natural gesture to the enactment of a tale, or in the words of Hollywood screenwriters, to *showing, not just telling*. It is to the practical implementation of this maxim with natural and expressive gestures that we now turn.

Embodied, Enactive Storytelling

The scientific community lacks agreement on a single cognitive model for explaining the processes of creative generation. The Brain Computer metaphor, most prominently described by (Putnam 1961), offers a premise that most cognitive engineers take for granted. This metaphor of Computationalism regards the brain as the underlying hardware, just like the hardware of a computer, and the mind as the software that runs on this physical platform. But to what extent does the hardware shape the software, or vice versa, and how is the synergy between mind and body achieved? While not attempting to resolve these vexing long-standing questions, this paper explores an intersection that is neglected in most modeling approaches. Collaborative enactive storytelling is the application of creative software that relies crucially on the physicality of its hardware, thus blurring the boundaries between external interactions and internal representations. So we adhere to a new theory in the philosophy of cognition, called Enactivism, that challenges Computationalism in suggesting that a precursor to high-level cognition is the dynamic interaction of an active organism within its physical environment. This environment is modeled internally not by translating sensory input into internal representations, but through exploratory interactions that create meaning (Di Paolo, Rohde, and De Jaegher 2010).

By analogy with these biological organisms, a robot can operate within, modify and learn from its environment (see, for instance, (Sandini, Metta, and Vernon 2007)). The robotic system becomes producer and product at the same time via the establishment of an *autopoietic feedback loop*. In such a feedback loop, the execution of a specific behavior by an actor for a spectator may trigger a corresponding response from the spectator, which in turn influences the actor's subsequent actions and shapes the actor's overall behavior (an explanation is given in (Fischer-Lichte 2012)). The most influential behavior of a storyteller is the use of the spoken word, but apt choices of physical gesture can also greatly contribute to the creation of an effective feedback loop (see (McNeill 1985), (Bergen, Narayan, and Feldman 2003)). As the interface between one's internal representations and the external environment, the body can make use of gestures to express what speech alone cannot convey. The following section outlines an implemented robot framework that provides two different modes of enactive storytelling.

Framework Description

The framework that is used for the two storytelling modes has been developed using *Python* and the *NAOqi* (Version 2.1.4.13) package. This software package provided by *Softbank Robotics* allows easy access to the different modalities of the robot's hardware. This section briefly describes the most important modules that have been used, databases that have been integrated for the story-generation process and the solution to problems that have been encountered during the setup of the framework. The databases used to craft the stories and the questions to create the interview-shaped mode will be made publicly available.

Automated Storytelling The framework is built upon the *Scéalextric* system for automated story-generation (Veale 2016a), as this provides a dense forest of plot possibilities for our robot to explore, perhaps in collaboration with a user. *Scéalextric*'s rich databases of symbolic representations allow actions to be bound together by causal connections and characters to be bound to actions on the basis of their established qualities. The system binds individual plot actions into plot segments (or arcs) with the following two-character triplet shape:

1. X *action* Y
2. Y *reaction* X
3. X *re-reaction* Y

The *Scéalextric* system provides over 3000 plot segments of this kind, made from causally-appropriate triplets that range over 800 different plot verbs (see (Veale 2017)). The resulting story space is modeled as a forest of trees in which each vertex is a plot verb, and in which every random walk yields a causally-coherent plot. Here is an example of a traversal through the forest:

A learns from B $\longrightarrow$ *A is inspired by B* $\longrightarrow$ *A falls in love with B* $\longrightarrow$ *A sleeps with B* $\longrightarrow$ *B fails to impress A* $\longrightarrow$ *A is disillusioned by B* $\longrightarrow$ *A breaks with B*

A robot story-teller can choose to explore the story forest using random walks, or it can elicit personal experience from the listener to guide its traversals. The precise strategy depends on which of its two modes the robot is operating within. In either case, the branching structure of the story forest provides choice (in random walk mode) and apposite junctures at which to question the user (in interview mode).

Gestures and body movement The current framework differs significantly from past efforts to exploit gestures on the Nao, drawing as it does from a set of 400+ predefined gestures from Aldebaran. We handcrafted annotations for each gesture, or physical behaviour, with one or more *Scéalextric* plot verbs. Each association of a gesture to a verb is also marked as strong, medium or weak according to our judgment (e.g., see Fig. 1). In all, 68% of the 800+ plot verbs in *Scéalextric* are associated with at least one gesture.

Understanding the Robot The current framework uses the Nao's *AnimatedSpeech* module to enrich its rendering of text as speech. Each sentence of a story is preprocessed prior its output to enhance its comprehensibility, e.g., by increasing of volume to the maximum, or lowering the voice pitch to simulate a more mature voice, or slowing the rate of articulation to increase understandability, or shortening of the pause between sentences to yield greater momentum in the telling. Unfortunately, the mechanical joints of a gesticulating Nao create noise that competes with the robot's speech, and in a non-laboratory environment this can impede story comprehension. We have thus introduced a super-title feature that echoes the output of the speech module on a large screen, so that an audience can follow the story in an additional modality (see e.g., Fig. 2).

Understanding the User The framework uses the Nao's *SpeechRecognition* module to communicate with the user. This module is primed with a vocabulary of words to which the robot should react via a built-in *word spotting* option. The vocabulary can be pre-loaded with thousands of words, yet the greater its size the more likely it is to confuse similar-sounding words. In some contexts we disable word spotting and require the user to reply with a single word response when explicitly prompted. As we shall see, such constraints need not impact the naturalness of a man-machine dialogue if the interactions are well-engineered and suitably framed.

Baseline Mode

The baseline mode of interactive storytelling employs minimum engagement with the user, but explores the same story forest and exploits the same gestural possibilities as other modes. Story-telling is initiated in this mode with a request

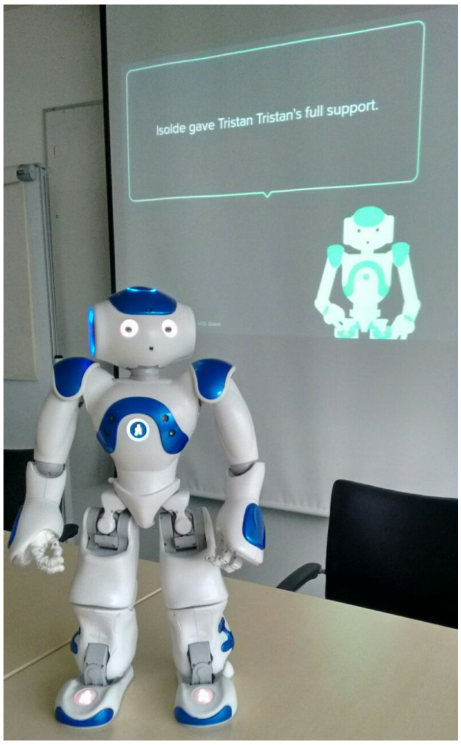


Figure 2: Demonstration of the storytelling robot with supertitles on the screen to increase comprehensibility.

to the user: "please provide an action around which to build a story." Any of *Scéalextric*'s 800+ action verbs may be offered in response by the user, as the Nao's speech recognizer is primed with the corresponding words. Low-level engineering challenges include loading this vocabulary in a parallel thread and preventing robotic stutters during the loading phase.

A vocabulary of 800+ verbs diminishes the reliability with which the robot can correctly distinguish words, as e.g. the words "look" may be confused for "cook" or "pay" for "pray." Fortunately the Nao's acoustic module reports a confidence value for each word that it recognizes. Only when this confidence is above an upper threshold (0.65) does the robot accept the user's response without question. When the confidence falls below a lower threshold (0.4) the robot remains in listening mode; only when the confidence falls between thresholds does the robot signal its uncertainty and seek explicit user confirmation with a "yes" or "no."

Once the robot has obtained an action from the user, story-generation around that pivot can proceed. Building on the *Flux Capacitor* representations of (Veale 2014), the robot selects its start and end points in the story forest to represent a meaningful character-development arc. Stories with the preferred number of actions (e.g., 6 to 10) are then generated by traversing the story forest between these end-points and retaining only those pathways/plots that contain the desired action. Of these matching pathways, the robot selects one that provides maximal opportunity for gestural expression, i.e. the story that contains the most verbs with gestural opportunities. Consider this complete example of a generated story using the keyword "*intimidate*":

- `Nao`: This story is about Isolde the Loyalist and Tristan the Despot.
- `Nao`: Isolde swore loyalty to trustworthy Tristan.
- `Nao`: Isolde gave Tristan Tristan's full support.
- `Nao`: But Tristan took full advantage of impressionable Isolde.
- `Nao`: Tristan intimidated others with threats of violence from Isolde.
- `Nao`: Thuggish Tristan threatened Isolde with violence.
- `Nao`: Isolde considered Tristan a disgusting monstrosity
- `Nao`: so Isolde's feelings for trustful Tristan soured.
- `Nao`: As a result, Isolde sold out Tristan to Tristan's enemies.

Each story begins with an introduction of a pair of characters, which have been selected from a comprehensive database of familiar faces, real and fictional, called the NOC list (see (Veale 2016b); access the list at `github.com/prosecconetwork`). The sequence of actions that have been selected from the story-graph are rendered with apt logical connectives (*and, so, but*, etc.) to convey the teller's understanding of causality.

In this baseline mode, the user is given only one opportunity – a single action – to influence the direction of a story. An example video of this mode can be found at: `bit.ly/2GGORhc`. More opportunities for collaborative development and self-expression are offered by the interview-shaped interaction of *Therapist mode*, as we'll see next.

Therapist Mode

We name this mode *Therapist mode* because it frames its interactions with the user around a series of probing questions about the user's own life experiences. These questions mirror the possibilities of the robot's story space because they are built from an interrogative rendering of the same plot actions. The guiding intuition is that everyone has a story in them; the role of a good therapist is to bring this tale to light by collaboratively reassembling the user's own experiences.

The story's actions should reflect the individual experiences while perhaps changing the sequence and filling gaps, thereby offering a new perspective on the user's past behavior. Potential story actions are rendered as yes-no questions (so, e.g., $are\_inspired\_by$ is rendered as "Have you ever been inspired by someone?"), and an affirmative answer adds that action to the story's plot. Consider this example:

1. `Nao`: Let us craft a story. I will start to ask you some questions. You only have to answer with yes or no.
2. `Nao`: Have you ever felt inspired by someone?
3. `User`: Yes.
4. `Nao`: But have you ever taken a hard look at someone who inspired you?
5. `User`: Correct.
6. `Nao`: Have you ever been criticized by that person?
7. `User`: No.
8. `Nao`: Can you imagine this is possible?
9. `User`: No.
10. `Nao`: Have you ever felt censured by that person?

The initial story verb is chosen at random from the robot's repertoire. If an affirmative response is given, the verb provides the next action in the story, and the potential follow-ons to that action suggest the next set of questions. Consequently, each of those interviews can vary in length, i.e. two or three positive replies might suffice to generate a story whereas negative answers will make the interview longer as the story tree needs to be explored further. If a negative response is given, and asserted again, the robot jumps to another branch in the story forest, and tries to move forward from there. We want the robot's stories to build on the user's experiences but to expand upon them too, to suggest what might have happened if events had taken a different turn. Thus, in line 6 the robot asks whether the user was ever criticized by a role model. Given a negative response, the robot presses on, asking instead if the user can conceive of this possibility. If the user now replies affirmatively, the story can incorporate this sequence of events (inspiration followed by criticism). To further allow for stories that go beyond the specific facts, users are also encouraged to reply with "maybe." As the questions asked of the user will differ from session to session, a different – yet highly personal – story will result each time.

Each story involves two characters, a protagonist (the user, or "you") and an antagonist. Notice how the questions above relate each action back to the previous action by assuming the antagonist to be common to both. It is likely that the user will have multiple antagonists in mind when answering questions, and the antagonist presumed in answer 3 is not the antagonist presumed by answer 13. The questions are sufficiently general to allow this artistic license to operate, so that the robot can weave stories that conflate several people from the user's life into a single thought-provoking antagonist. Once the session ends at the user's request, the selected actions can now be woven into a two-character plot:

> This is the story about you and a pioneer. This spectacular pioneer became a shining inspiration for you. You kept the pioneer under close observation. You mimicked the popular pioneer's style and adopted it as your own. "You've let me down" said the Pioneer plaintively, so the domineering pioneer gave you a very public rap on the knuckles, and to conclude, the pioneer brought suit against you in open court. That's the end of your story.

This story features all of the actions that the user has assented to, and may include additional actions as well to conclude the protagonist's arc (in the sense of (Veale 2014)) and bring the tale to a satisfying conclusion. Notably, the protagonist of the story is addressed as 'you', while the protago-

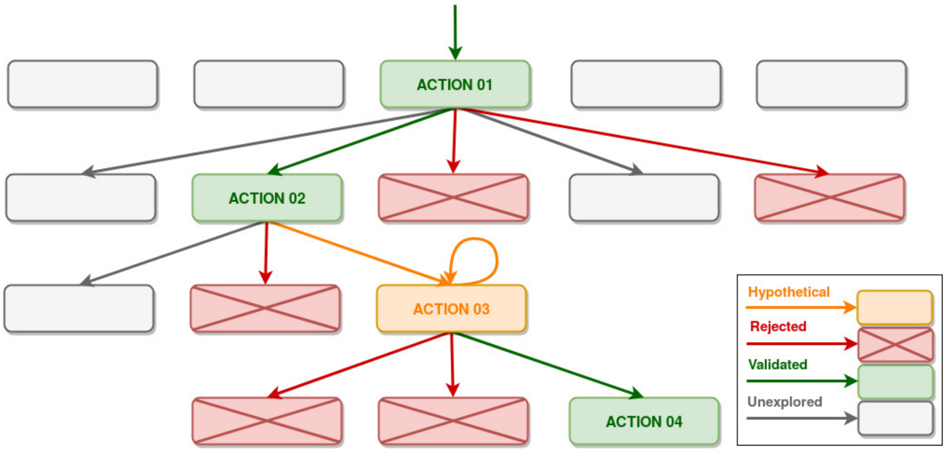


Figure 3: Example of a tree traversal for the story generation process. Grey nodes and edges indicate that a path has not been explored by the system. Red nodes and edges denotes those that have been answered with a 'No' by the user. Green nodes have been validated by the user, giving the sequence *ACTION 01*, *ACTION 02*, *ACTION 03* and *ACTION 04* that start the story story. A yellow node is one that has been rejected by a direct question, but accepted by the user to be a hypothetical possibility.

nist is referred to not by name but by character-type (here, the *pioneer*, as this is the kind of character that *inspires* others). The resulting story diverges from the user's own experiences, but in doing so sheds new light on them.

A schematic view of the generation-by-interrogation process is provided in Figure 3. Here we can see that the first selected action (`ACTION 01`) is translated into a question form and posed to the user by the robot. The node is selected (green) if the user validates this experience, or assents to the hypothetical, and this selection cues the node's children as possible follow-ons. However, if the user negates the question *and* its rephrasing as a hypothetical, the node and its action are blacklisted so that it cannot be asked again of the user. Instead, the robot selects another sibling and another branch to explore. In the example the node `ACTION 03` has been validated as a meaningful hypothetical by the user.

This mode differs significantly from the baseline in the amount of interaction it demands from the user. However, this interaction is not deterministic, and the user's answers merely suggest, rather than dictate, the robot's path through the story forest. This mode demonstrates that every story can offer a probing interrogation of one's own experiences, and vice versa whilst creating a feedback loop. An example video of this mode can be found at: `bit.ly/2oouZbY`. Our work up to this point has focused on development of the pilot system, but in the next section we comment on the planned evaluation of both modes.

Evaluation Challenges A crowd-based evaluation of the *Scéalextric* model of story-generation has been reported in (Veale 2017). In that work, two modes of story-generation were evaluated. Each mode employed the same plotting mechanism – a traversal of the story forest and subsequent rendering of plot actions into connected linguistic utterances – but each relied on a different model of characterization. In the generic condition, plots were instantiated with animal protagonists and antagonists, such as "the monkey" and "the snake," that were chosen at random from a list of Aesop-style animals. In the more elaborate familiarity condition, established characters such as Darth Vader or Donald Trump were chosen from the NOC list, and aspects of their characters as retrieved from the NOC were integrated into the rendering of plot verbs. Moreover, characters were chosen randomly in apt pairs, so that stories would pit Donald Trump against Lex Luthor, or Darth Vader against Bane, or Steve Jobs against Leonardo Da Vinci. The crowd-based evaluation of fifty stories from each mode solicited 10 ratings for six dimensions for each story, and the NOC-based mode showed superior results for all dimensions. Surprisingly, the NOC stories scored higher for dramatic content too, even though the underlying plots relied on precisely the same plotting mechanism as the generic stories. Consequently, we must take this bias into account comparing the two modes and eventually disable the NOC characters in order to compare the baseline with the therapist mode.

Once again we find ourselves with two modes of story-generation to evaluate. In the earlier crowd-based evaluation of (Veale 2017), judges were presented with pre-generated stories and asked to rate them after-the-fact. However, as the current stories are generated in cooperation with the user, and rely crucially on the user's input (as well as an appreciation of their own past actions), these stories are far less amenable to a simple crowd-based evaluation. Moreover, our evaluation should allow us to test the central research question of the current work:

> To what extent can an interview-style collaboration enrich the storytelling experience between human and machine over an approach with minimal interaction?

Since we further hypothesize that the embodied, gestural behaviour of the robot will form a significant part of any such enrichment, it behooves us to present the robot's stories *live*, in the physical presence of the robot. While video-taped sessions are one possibility when the goal is to evaluate the impact of gestures, prerecorded video does not support real-time interactivity *and* gesture in the same test. As the system moves out of the pilot stage, we hope to evaluate the system with a live audience, most likely in an educational setting where participants can be given partial credit for their feedback. For now, we aim to gain practical insights into possible evaluations in show-and-tell sessions at conferences.

Philosophical Investigations

Symbol-grounding remains a vexing problem for modern AI systems, and especially for those that rely on wholly symbolic representations. How do the symbols of a representation relate to the things in the world for which they are supposed to stand (Searle 1980)? So, for instance, how does the representation of a given plot verb in *Scéalextric* relate to the intuitive understanding of this verb as held by the human audience for a story that uses this verb? In a weak sense, the elements of a symbolic representation can mutually ground each other if they are connected in inferentially useful ways. Thus, insofar as the verbs in *Scéalextric*'s story forest are connected to each other in causally significant ways that are appropriately marked, the symbols for these verbs are weakly grounded in an extensional model of causality.

As system builders, our goal is not a theoretical grounding of symbols but the practical use of symbols in ways that *appear* grounded, whatever the philosophical truth may be. In this respect, the embodiment of a story-telling system in a physical agent capable of nuanced gestures goes a long way toward selling the appearance of grounding. But in addition to physical grounding, it is also meaningful to speak of *psychological* grounding. Does a system employ its cultural symbols in ways that respect the psychological attitudes that native speakers ascribe to those symbols? This is a question that goes beyond purely plot-related issues. Rather, it goes to how humans "feel" about specific symbols. We argue that framing the story-generation process as a therapeutic interview is just as effective at achieving psychological grounding as it is at achieving meaningful man-machine collaboration. Ultimately, the robot does not succeed in grounding its symbols in any way that would satisfy (Searle 1980), but it does come close to reflecting the grounding possessed by the human audience, whatever that might be.

Being There: Identity and Empathy in Storytelling

A study by (Pop et al. 2013) has demonstrated that the use of a social robot to support interaction in story-telling is more effective than the use of text on a screen. For when identification with the teller is the goal, the physical presence of a moving body with a human shape makes all the difference. While reading text on the screen is a solitary activity, a shared gaze with two human-like eyes can keep us focused and engaged during an interaction. Moreover, a study by (Seo et al. 2015) has shown that people tend toward greater empathy with a physical robot than with a simulated on-screen avatar. This heightens the contribution of physical modalities such as gesture, gaze and speech to the identification of listener with speaker and of human with machine. A listener that can identify with the storyteller is better positioned to empathize with the story that the teller wants to convey, especially when that story is crafted from the life experiences of the listeners themselves. Future studies are needed to investigate the empirical effect of a physically embodied and psychologically-grounded robot story teller that invents and delivers its *own* stories to users. While it is still at an early stage of development, we believe that *Scéalextric-NAO* is an important step in the right direction.

Acknowledgments

We would like to thank Stefan Riegl for his contribution to implementation of the storytelling system as outlined here.

References

Ackerman, M., and Loker, D. 2017. Algorithmic songwriting with alysia. In *International Conference on Evolutionary and Biologically Inspired Music and Art*, 1–16. Springer.

Bergen, B.; Narayan, S.; and Feldman, J. 2003. Embodied verbal semantics: Evidence from an image-verb matching task. In *Proceedings of the Twenty-Fifth Annual Conference of the Cognitive Science Society*, 139–144.

Chen, G.-D.; Nurkhamid; and Wang, C.-Y. e. a. 2011. A survey on storytelling with robots. In *International Conference on Technologies for E-Learning and Digital Entertainment*, 450–456. Springer.

Concepción, E.; Gervás, P.; and Méndez, G. 2017. An api-based approach to co-creation in automatic storytelling. In *6th International Workshop on Computational Creativity, Concept Invention, and General Intelligence. C3GI.*

Csapo, A.; Gilmartin, E.; Grizou, J.; Han, J.; Meena, R.; Anastasiou, D.; Jokinen, K.; and Wilcock, G. 2012. Multimodal conversational interaction with a humanoid robot. In *Cognitive Infocommunications (CogInfoCom), 2012 IEEE 3rd International Conference on*, 667–672. IEEE.

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. ACM.

Di Paolo, E.; Rohde, M.; and De Jaegher, H. 2010. Horizons for the enactive mind: Values, social interaction, and play. *Enaction: Towards a new paradigm for cognitive science*.

Fischer-Lichte, E. 2012. *Ästhetik des Performativen*. Suhrkamp Verlag.

Gelin, R.; d'Alessandro, C.; Le, Q. A.; Deroo, O.; Doukhan, D.; Martin, J.; Pelachaud, C.; Rilliard, A.; and Rosset, S. 2010. Towards a storytelling humanoid robot. In *AAAI Fall Symposium: Dialog with Robots*.

Gervás, P. 2013. Propp's morphology of the folk tale as a grammar for generation. In *OASIcs-OpenAccess Series in Informatics*, volume 32. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik.

Gouaillier, D.; Hugel, V.; Blazevic, P.; Kilner, C.; Monceaux, J.; Lafourcade, P.; Marnier, B.; Serre, J.; and Maisonnier, B. 2009. Mechatronic design of nao humanoid. In

Robotics and Automation, 2009. ICRA'09. IEEE International Conference on, 769–774. IEEE.

Ham, J.; Bokhorst, R.; Cuijpers, R.; van der Pol, D.; and Cabibihan, J. 2011. Making robots persuasive: the influence of combining persuasive strategies (gazing and gestures) by a storytelling robot on its persuasive power. In *International conference on social robotics*, 71–83. Springer.

Häring, M.; Bee, N.; and André, E. 2011. Creation and evaluation of emotion expression with body movement, sound and eye color for humanoid robots. In *Ro-Man, 2011 Ieee*, 204–209. IEEE.

Hood, D.; Lemaignan, S.; and Dillenbourg, P. 2015. When children teach a robot to write: An autonomous teachable humanoid which uses simulated handwriting. In *Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction*, 83–90. ACM.

Jordanous, A. 2017. Co-creativity and perceptions of computational agents in co-creativity. In *Proceedings of the Eighth International Conference on Computational Creativity, Atlanta, US. ACC*.

Krebs, D. 1975. Empathy and altruism. *Journal of personality and social psychology* 32(6):1134.

Le, Q. A., and Pelachaud, C. 2012. Evaluating an expressive gesture model for a humanoid robot: Experimental results. In *Submitted to 8th ACM/IEEE International Conference on Human-Robot Interaction*.

Le, Q. A.; Hanoune, S.; and Pelachaud, C. 2011. Design and implementation of an expressive gesture model for a humanoid robot. In *Humanoid Robots (Humanoids), 2011 11th IEEE-RAS International Conference on*, 134–140. IEEE.

McNeill, D. 1985. So you think gestures are nonverbal? *Psychological review* 92(3):350.

Meena, R.; Jokinen, K.; and Wilcock, G. 2012. Integration of gestures and speech in human-robot interaction. In *Cognitive Infocommunications (CogInfoCom), 2012 IEEE 3rd International Conference on*, 673–678. IEEE.

Méndez, G.; Gervás, P.; and León, C. 2016. On the use of character affinities for story plot generation. In *Knowledge, Information and Creativity Support Systems*. Springer. 211–225.

Parde, N.; Hair, A.; Papakostas, M.; Tsiakas, K.; Dagioglou, M.; Karkaletsis, V.; and Nielsen, R. D. 2015. Grounding the meaning of words through vision and interactive gameplay. In *IJCAI*, 1895–1901.

Pelachaud, C.; Gelin, R.; Martin, J.; and Le, Q. A. 2010. Expressive gestures displayed by a humanoid robot during a storytelling application. *New Frontiers in Human-Robot Interaction (AISB), Leicester, GB*.

Pérez y Pérez, R., and Sharples, M. 2001. Mexica: A computer model of a cognitive account of creative writing. *Journal of Experimental & Theoretical Artificial Intelligence* 13(2):119–139.

Pop, C. A.; Simut, R.; Pintea, S.; Saldien, J.; Rusu, A.; Vanderfaeillie, J.; David, D.; Lefeber, D.; and Vanderborght, B. 2013. Social robots vs. computer display: does the way social stories are delivered make a difference for their effectiveness on asd children? *Journal of Educational Computing Research* 49(3):381–401.

Putnam, H. 1961. Brain and behavior, originally read as part of the program of the american association for the advancement of science. *Section L (History and Philosophy of Science)*.

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

Sandini, G.; Metta, G.; and Vernon, D. 2007. The icub cognitive humanoid robot: An open-system research platform for enactive cognition. In *50 years of artificial intelligence*. Springer. 358–369.

Searle, J. 1980. Minds, brains, and programs. *Behavioral and brain sciences* 3(3):417–424.

Seo, S.; Geiskkovitch, D.; Nakane, M.; King, C.; and Young, J. 2015. Poor thing! would you feel sorry for a simulated robot?: A comparison of empathy toward a physical and a simulated robot. In *Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction*, 125–132. ACM.

Shamsuddin, S.; Ismail, L.; Yussof, H.; Zahari, N.; Bahari, S.; Hashim, H.; and Jaffar, A. 2011. Humanoid robot nao: Review of control and motion exploration. In *Control System, Computing and Engineering (ICCSCE), 2011 IEEE International Conference on*, 511–516. IEEE.

Singh, D.; Ackerman, M.; and Pérez y Pérez, R. 2017. A ballad of the mexicas: Automated lyrical narrative writing. In *Eighth International Conference on Computational Creativity, ICCC, Atlanta*.

Tapus, A.; Peca, A.; Aly, A.; Pop, C.; Jisa, L.and Pintea, S.; Rusu, A.; and David, D. 2012. Children with autism social engagement in interaction with nao, an imitative robot: A series of single case experiments. *Interaction studies* 13(3):315–347.

Veale, T. 2013. A service-oriented architecture for computational creativity. *Journal of Computing Science and Engineering* 7(3):159–167.

Veale, T. 2014. Coming good and breaking bad: Generating transformative character arcs for use in compelling stories. In *Proceedings of ICCC-2014, the 5th International Conference on Computational Creativity*.

Veale, T. 2016a. A rap on the knuckles and a twist in the tale. *AAAI spring symposium series*.

Veale, T. 2016b. Round up the usual suspects: Knowledge-based metaphor generation. In *Proceedings of the Fourth Workshop on Metaphor in NLP*, 34–41.

Veale, T. 2017. Déjà vu all over again: On the creative value of familiar elements in the telling of original tales. In *Proceedings of ICCC-2017, the 8th International Conference on Computational Creativity*, 245–252.

Wilcock, G., and Jokinen, K. 2013. Wikitalk human-robot interactions. In *Proceedings of the 15th ACM on International conference on multimodal interaction*, 73–74. ACM.

CAEMSI : A Cross-Domain Analytic Evaluation Methodology for Style Imitation

Jeff Ens, Philippe Pasquier
School of Interactive Arts and Sciences
Simon Fraser University
{jeffe,pasquier}@sfu.ca

Abstract

We propose CAEMSI, a cross-domain analytic evaluation methodology for Style Imitation (SI) systems, based on a set of statistical significance tests that allow hypotheses comparing two corpora to be tested. Typically, SI systems are evaluated using human participants, however, this type of approach has several weaknesses. For humans to provide reliable assessments of an SI system, they must possess a sufficient degree of domain knowledge, which can place significant limitations on the pool of participants. Furthermore, both human bias against computer-generated artifacts, and the variability of participants' assessments call the reliability of the results into question. Most importantly, the use of human participants places limitations on the number of generated artifacts and SI systems which can be feasibly evaluated. Directly motivated by these shortcomings, CAEMSI provides a robust and scalable approach to the evaluation problem. Normalized Compression Distance, a domain-independent distance metric, is used to measure the distance between individual artifacts within a corpus. The difference between corpora is measured using test statistics derived from these inter-artifact distances, and permutation testing is used to determine the significance of the difference. We provide empirical evidence validating the statistical significance tests, using datasets from two distinct domains.

Introduction

There is growing demand for creative generative systems in the entertainment industry, which has prompted an abundance of research in the area of Style Imitation (SI). Given a corpus $C = \{c_1, ..., c_n\}$, SI systems aim to generate new artifacts that emulate the stylistic characteristics of C. Many of these SI systems generate some form of musical content, including; harmonic progressions, melodies (Yang, Chou, and Yang 2017), and polyphonic compositions (Liang et al. 2017). A more comprehensive overview of the work in the domain can be found elsewhere (Pasquier et al. 2017; Briot and Pachet 2017). In the visual art domain, the Creative Adversarial Network (CAN) is trained to generate visual art that deviates from the styles it has already learned (Elgammal et al. 2017). Moreover, many Natural Language Generation (NLG) systems have been developed that generate jokes, poetry, and narratives in a particular style (Gatt and Krahmer 2017). To accommodate the large influx of generative systems in recent years, we propose CAEMSI [1].

Ritchie mentions two conditions for determining if creativity has occurred: *novelty*, the degree to which an artifact is dissimilar to other examples within the corpus and *quality* (Ritchie 2007). He also emphasizes the notion of *typicality*, the degree to which a generated artifact is representative of the source corpus (C). In the context of style imitation, measuring typicality is of critical importance, as the performance of an SI system hinges on its ability to emulate the stylistic characteristics of the source corpus. As a result, CAEMSI focuses on measuring the typicality of a generated corpus, with respect to the source corpus. Although novelty is also an important indicator of the system's quality, as it is generally undesireable for an SI system to plagiarize large sections from the source corpus, we leave this aspect of evaluation for future work.

Traditionally, participants assess the capacity of a particular system to emulate a particular style, allowing researchers to make claims about the success of that system. Unfortunately, this is not a scalable solution, and can make it difficult to compare SI systems. With the long-term goal of creating highly capable SI systems, it is necessary to develop robust methods for the evaluation of these systems, as a lack of methodical evaluation can have a negative effect on research progress (Pearce, Meredith, and Wiggins 2002). The approach described in this paper is domain independent, harnessing the power of Normalized Compression Distance (NCD) (Cilibrasi and Vitányi 2005) and permutation testing to provide a scalable solution to the problem of SI system evaluation. In order to demonstrate the effectiveness of this approach, we conduct experiments on datasets in two different domains; the Wikiart image dataset[2] and the Classical Archives MIDI dataset[3].

Background

Evaluation Methodologies

Although many methodologies that evaluate the creative capacity of a generative system have been proposed, we will limit our discussion to those which have been used to measure typicality. In general, we can divide these methodolo-

[1] The code is available https://goo.gl/ejN1RM

[2] https://www.wikiart.org/

[3] https://www.classicalarchives.com/midi.html

gies into two categories, those which rely on human participants, and those based purely on computation. Unto our knowledge, the only statistical evaluation methodology for typicality was proposed by Gonzalez Thomas et al., however it is only capable of evaluating melodic composition systems (Gonzalez Thomas et al. 2013).

The Consensual Assessment Technique (CAT) (Amabile 1982) is based on the notion that experts are the most capable of distinguishing creative artifacts within their respective domain. To account for discrepancies, which arise given the subjective nature of these assessments, the CAT averages the assessments of several experts. Pearce and Wiggins employ the CAT to evaluate the success of melodic generation algorithms (Pearce and Wiggins 2007).

Another approach, inspired by the Turing-test, measures participants ability to discriminate between computer-generated artifacts and artifacts from the source corpus. This evaluation methodology has been used to evaluate many SI systems, including a Deep LSTM Network that generates Bach chorales (Liang et al. 2017), and a Generative Adversarial Network that generates images (Elgammal et al. 2017).

Related Work

To the best of the authors' knowledge, there are no domain independent metrics for typicality, however, several quantitative metrics for creative systems have been proposed. Maher has proposed two metrics for measuring creativity quantitatively. The first, equates *novelty* with distance from predominant clusters of artifacts, measures *surprise* using pattern matching algorithms, and calculates *value* using a fitness function (2010). However, it is not clear how the proposed metrics would be applied to an arbitrary domain, and no proof of concept is provided. The second, uses Bayesian inference to measure the novelty of an artifact, which is used to evaluate potential designs for laptop computers (Maher and Fisher 2012).

Burns measures creativity as the combination of psychological arousal, which is computed using Shannon entropy, and appraisal, which is computed using Bayesian theory (Burns 2015). The Regent-Dependent Creativity (RDC) metric measures value and novelty. Artifacts are represented by a set of pairs $(P(\textit{regent}, \textit{dependent}))$, where *regent* is an action or attribute, and *dependent* is a state or target for an action (Rocha, Ribeiro, and El 2016). Using a graph, which includes associations between artifacts, they propose metrics to measure synergy, the value produced by various elements acting cooperatively, and Bayesian surprise, the degree to which an artifact is unexpected or novel. Although this metric seems to work well for the low dimensional problems presented in the paper, it is not clear that this approach could efficiently handle artifacts which require a large number of pairs for representation. Furthermore, it relies on the domain knowledge of synergy, which is difficult to determine in some domains.

Motivation

Although human-based evaluation methodologies are not without their strengths, the shortcomings of these methodologies directly motivated the development of the statistical tests proposed in this paper.

Domain Knowledge

Accurately assessing the typicality of an artifact with respect to a source corpus, requires a significant amount of domain knowledge, as the participant must be familiar with the stylistic characteristics of the source corpus. This issue is exacerbated when performing a CAT, since participants must have an expert level knowledge of the source corpus. Undoubtedly, this is one of the primary reasons an abundance of musical SI systems have focused on imitating Bach chorales, as there is a large pool of experts, and most people are familiar with Bach's work. Since a lack of domain knowledge undermines the reliability of the evaluation process, the types of scientific inquiries which have been explored are biased by restrictions on the source corpora, placing limitations on scientific progress in this area.

Bias Against Generative Systems

Previous research has shown that when participants were asked to distinguish between two folk melodies, some of which were human-composed and others which were recombinations of the human-composed melodies, participants attributed unusual or disagreeable human compositions to the computer (Dahlig and Schaffrath 1997). Norton, Heath, and Ventura found a significant bias against images labeled as being generated by a computer (2015). In contrast, several studies have demonstrated that the knowledge that a computer created a piece of music, does not significantly affect the participants' evaluation and enjoyment of the piece (Moffat and Kelly 2006; Friedman and Taylor 2014; Pasquier et al. 2016). Although Moffat and Kelly's study did not explicity test the same hypothesis as Dahlig and Schaffrath, their results corroborate the same conclusion, as participants attributed compositions they disliked to the computer, independent of their actual authorship.

When participants are tasked with making the distinction between human-generated and computer-generated artworks, they may in fact be searching for features which they expect to be generated by a computer, rather than focusing on the broader style of the composition (Ariza 2009). As a result, the test degenerates to one which is focused on counting perceived mistakes. This issue has been highlighted by Pearce in his discussion on the evaluation of musical composition systems (2005). Clearly, this type of bias is very problematic when attempting to evaluate an SI system that imitates artifacts that humans tend to find disagreeable, such as the atonal works of Arnold Schoenberg.

Variability

The subjective nature of creativity-based assessments poses problems for the systematic evaluation of creative systems in general. There is evidence that cultural background can have an effect on how an artifact is perceived. For example, Eerola et al. found that western and African listeners perceived musical attributes differently (2006). Furthermore, environmental factors will affect the reliability of these assessments, including the equipment used to observe the artifact, and the physical condition of the participant. Although those who design experiments take many steps to mitigate the effects of these factors, Schedl at al. (Schedl, Flexer, and Urbano 2013) provide evidence that inter-rater agreement is still limited in a practical setting. In one case, non-experts' assessments of poetry were found to be negatively correlated with the assessments of experts (Lamb, Brown, and Clarke 2015). Similarly, Kaufman, Baer, and Cole found that experts were far more reliable than non-experts, when asked to judge the creativity of a short story, as measured by inter-rater reliability for both groups (2009).

Scalability

Unfortunately, using human participants places limitations on the total number of assessments that can be collected. Participants are only capable of making so many assessments before fatigue will begin to degrade the quality of their responses. Notably, this problem is exacerbated by the limited number of participants involved when conducting a *CAT*. Although crowdsourcing does make it easier to collect a large number of assessments, there are still monetary and time limitations that place restrictions on the the total number of assessments that can be feasibly collected. Clearly, the limited scalability of these evaluation methods is in direct conflict with the large number of artifacts which generative systems can produce.

In many cases, a small subset of the generated artifacts is used to evaluate the system, decreasing the number of assessments required. However, issues will naturally arise when the selected subset is not adequately representative of the system's output as a whole (Ariza 2009). Moreover, it is not trivial to determine if a subset of artifacts is representative of the systems output a priori. Most importantly, these limitations make it increasingly difficult to evaluate a large number of systems.

The Proposed Solution

In contrast to human-based evaluation methods, CAEMSI eschews the issues of domain knowledge, human bias, and variability. Admittedly, there are still limitations with respect to the size of corpora, which will be addressed in future work. However, computation based methods of evaluation are far more scalable than human-based solutions, as computers can process artifacts much faster than humans can.

Statistical Tests for Typicality

In what follows, $X = [x_i, i = 1, ..., n]$ denotes a vector X, containing n elements. $X \oplus Y$ denotes the concatenation of two vectors. We use the term *corpora* to denote a vector of binary strings. $\mu(X)$ denotes the mean of a vector X, while $\phi(X)$ denotes the median. p_{diff} and p_{eqv} denote the significance of the statistical test for difference and equivalence respectively.

Given two corpora, $A = [a_i, i = 1, ..., n]$ and $B = [b_i, i = 1, ..., m]$, we test the null hypothesis $H_{D0} : A = B$ ($p_{\text{diff}} > \alpha$) against $H_{D1} : A \neq B$ ($p_{\text{diff}} \leq \alpha$) and the null hypothesis $H_{E0} : A \neq B$ ($p_{\text{eqv}} > \alpha$) against $H_{E1} : A = B$ ($p_{\text{eqv}} \leq \alpha$). When the result of a statistical test is insignificant, we accept the null hypothesis, which only indicates that there was insufficient evidence to support the alternate hypothesis, and does not validate or invalidate the null hypothesis. As a result, accepting the null hypothesis $H_{D0} : A = B$ is not the same as rejecting the null hypothesis $H_{E0} : A \neq B$ and accepting the alternative hypothesis $H_{E1} : A = B$, as only the latter indicates that $A = B$. Consequently, we can determine if $A = B$ using p_{eqv} and if $A \neq B$ using p_{diff}.

Normalized Compression Distance

Put simply, the *Kolmogorov complexity* $(K(x))$ of a finite length binary string x is the minimum number of bits required to store x without any loss of information. More formally, $K(x)$ denotes the length of the shortest Universal Turing Machine that prints x and stops (Solomonoff 1964). Intuitively, the minimum number of bits required to store a random string would be close to the number of bits used to represent the original string. As a result, a random string would have a high Kolmogorov complexity. In contrast, a string with a large number of repeated subsequences, would have a low Kolmogorov complexity. Although Kolmogorov complexity provides an absolute lower bound on the compression of a string, $K(x)$ is non-computable (Li et al. 2004), so a real-world compressor is used to approximate $K(x)$ in practice.

The *conditional Kolmogorov complexity* $(K(x|y))$ of a string x relative to a string y, denotes the length of the shortest program that prints x and stops, with y provided as additional input to the computational process. For example, if $x \simeq y$, $K(x|y)$ would be very small, as the program could reproduce x from y without requiring much additional information. In contrast, if x and y are highly dissimilar, $K(x|y)$ would be quite large.

Information distance is the length of the shortest binary program that can compute x from y and y from x. As a result, when x and y have a lot of mutual information, the length of this program will be fairly short. Li et al. propose the *normalized information distance* (1).

$$d(x, y) = \frac{\max(K(x|y), K(y|x))}{\max(K(x), K(y))} \tag{1}$$

Since $K(x|y) \simeq K(xy) - K(x)$ (Li et al. 2004), where xy denotes the concatenation of strings x and y, we can reformulate (1) to arrive at a computable *normalized compression distance* (NCD) (2). In practice, $K(x)$ is the length of string produced by a real-world compression algorithm, such as zlib. Although we tested several compression algorithms, we did not notice significant variation in terms of performance.

$$D(x, y) = \frac{K(xy) - \min(K(x), K(y))}{\max(K(x), K(y))} \tag{2}$$

Li et al. demonstrate that *NCD* is a universal distance metric, satisfying the following constraints.

1. $D(x, y) = 0$ *iff* $x = y$ (Identity)
2. $D(x, y) + D(y, z) \geq D(x, z)$ (Triangle Equality)
3. $D(x, y) = D(y, x)$ (Symmetry)

Notably, *NCD* has been applied to problems in a variety of domains, including music classification (Cilibrasi, Vitányi, and De Wolf 2004; Li and Sleep 2005), protein sequence classification (Kocsor et al. 2006), image registration (Bardera et al. 2010), and document classification (Axelsson 2010). Väyrynen and Tapiovaara use *NCD* to evaluate *machine translation* (MT) by measuring the distance between the predicted translation and the ground truth translation (Väyrynen and Tapiovaara 2010).

Distance Matrix Construction

Given a valid distance metric D and two corpora ($A = [a_i, i = 1, ..., n]$ and $B = [b_i, i = 1, ..., m]$), we can construct a pairwise distance matrix M, where $M_{ij} = D(c_i, c_j)$, and $C = A \oplus B = [c_i, i = 1, ..., n+m]$. We use several subsets of M to perform the proposed statistical tests. In the formula below, w_A and w_B are vectors containing all distinct within group distances for corpora A, and B respectively, while $b_{A,B}$ contains all between group distances. Notably, $l = n + m$ in the equations below.

$$w_A = [M_{ij}, i = 1, ..., n; j = 1, ..., n; j > i] \quad (3)$$

$$w_B = [M_{ij}, i = n+1, ..., l; j = n+1, ..., l; j > i] \quad (4)$$

$$b_{A,B} = [M_{ij}, i = 1, ..., n; j = n+1, ..., l] \quad (5)$$

Permutation Testing

A *permutation test* is a statistical significance test which requires no prior knowledge about the distribution of the test statistic under the null hypothesis, as this distribution is generated by calculating the test statistic for each possible labelling of the data. For example, consider the vector $C = A \oplus B$, which is comprised of two corpora delineated by the labels $\mathbf{L} = [l_i, i = 1, ..., n+m; l_{i \leq n} = 0, l_{i > n} = 1]$, and a test statistic $S = \mu(C_0) - \mu(C_1)$, where $C_j = \{c_i \mid l_i = j\}$. First, compute S using $\mathbf{L}$. Then compute S for each possible permutation of $\mathbf{L}$ to construct the distribution under the null hypothesis. Since the number of permutations grows exponentially, as comparing two corpora of size 50 would require $\binom{100}{50} \simeq 10^{29}$ distinct permutations, we approximate this procedure by randomly selecting m permutations. This procedure accommodates complex test statistics, for which it would be intractable, or overly difficult, to compute the distribution of the test statistic under the null hypothesis.

Testing for Difference

To test the hypothesis that two corpora are different, we adapt a permutation testing framework that was used to compare two groups of brain networks (Simpson et al. 2013). Simpson et al. create a pairwise distance matrix M using the Kolmogorov-Smirnov statistic, however, we use *NCD* instead.

$$R(M) = \frac{\mu(b_{A,B})}{\mu(w_A \oplus w_B)} \quad (6)$$

When R is greater than 1, the average between group distance is greater than the within group distance. Therefore, $R > 1$ suggests that the two corpora are likely distinct. In contrast, when $R \simeq 1$, there is likely no difference between the two corpora. The proposed test is detailed in the steps below, where $\mathbf{I}(\cdot) = 1$ if $(\cdot)$ is true and 0 otherwise.

1. Given two corpora $A = [a_i, i = 1, ..., n]$ and $B = [b_i, i = 1, ..., m]$, create a pairwise distance matrix M using (2).
2. Calculate the test statistic $T = R(M)$ using (6).
3. Take a random permutation (u^*) of the ordering $u = (1, ..., n+m)$ and reorder the columns and rows using this ordering to create M^*.
4. Calculate the test statistic $T^* = R(M^*)$ using (6).
5. Repeat steps 3 and 4 N times, producing the output $[T^*_n, n = 1, ..., N]$.
6. Calculate the p-value, $p_{\text{diff}} = \sum_{n=1}^{N} \mathbf{I}(T^*_n \geq T)/N$.

Testing for Equivalence

The proposed test for the equivalence of two corpora, is based on the following assumption.

$$(w_A = b_{A,B}) \wedge (w_B = b_{A,B}) \implies A = B \quad (7)$$

The intuition behind this assumption is shown in Figure 1 and 2, which show the cumulative distributions of w_A, w_B, and $b_{A,B}$ for an intra-artist comparison and an inter-artist comparison respectively. When two distinct corpora are compared, $b_{A,B} \neq w_A$ and $b_{A,B} \neq w_B$, as shown in Figure 1. In contrast, when two similar corpora are compared, $b_{A,B} \simeq w_A \simeq w_B$, as shown in Figure 2. In practice, the distributions of w_A, w_B, and $b_{A,B}$ are frequently skewed, and sometimes multi-modal, which necessitates a non-parametric test for equivalence.

As a result, we employ a permutation testing framework (Pesarin et al. 2016), which is based on Roy's Union-Intersection approach (1953), to test for the equivalence of two distributions. First, it is necessary to define an equivalence interval on which the two distributions will be considered equal. ε_I and ε_S denote the inferior and superior margins, respectively. Then we test two hypotheses; $H_{I0} : \delta \geq -\varepsilon_I$ against $H_{I1} : \delta < -\varepsilon_I$ and $H_{S0} : \delta \leq \varepsilon_S$ against $H_{S1} : \delta > \varepsilon_S$, where δ is the divergence between the two distributions being compared. In some cases, this is measured as the difference between the means (μ), however we use the difference between the medians (ϕ), as it is more robust to outliers. As a result, the global null hypothesis (H_{E0}) is true if both one-sided null hypotheses (H_{I0}, H_{S0}) are true, and the global alternative hypothesis (H_{E1}) is true if at least one of H_{I1} and H_{S1} is true. The following algorithm is used to test for the equivalence of two distributions.

1. Given two vectors $F = [f_i, i = 1, ..., n]$ and $G = [g_i, i = 1, ..., m]$, compute the rank transform of $F \oplus G$ to derive a rank transformed version F and G.
2. Given the superior and inferior equivalence margins $(\varepsilon_I, \varepsilon_S)$, we create two vectors $X_I = F \oplus (G + \varepsilon_I)$ and $X_S = F \oplus (G - \varepsilon_S)$, and an ordering $u = (1, ..., n+m)$.
3. Compute the test statistic for both hypothesis $T_I = \phi(X_{IF}) - \phi(X_{IG})$ and $T_S = \phi(X_{SG}) - \phi(X_{SF})$ where

$$X_{IF} = [X_I(u_i), i = 1, ..., n]$$
$$X_{IG} = [X_I(u_i), i = n+1, ..., n+m]$$
$$X_{SF} = [X_S(u_i), i = 1, ..., n]$$
$$X_{SG} = [X_S(u_i), i = n+1, ..., n+m]$$

   and $X(j)$ denotes the j_{th} element in X.
4. Take a random permutation (u^*) of the ordering u.
5. Compute the test statistics using the ordering u^*. $T^*_I = \phi(X_{IF}) - \phi(X_{IG})$ and $T^*_S = \phi(X_{SG}) - \phi(X_{SF})$.
6. Repeat steps 3 and 4 N times to simulate the distribution of the two partial test statistics, producing the output $[(T^*_{In}, T^*_{Sn}), n = 1, ..., N]$.

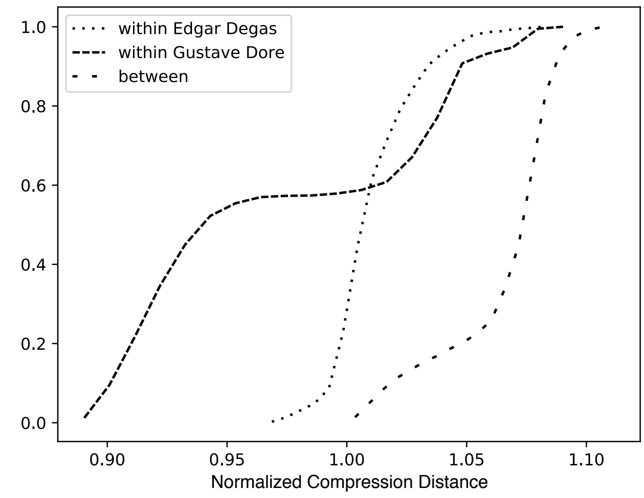


Figure 1: The cumulative *NCD* distributions (w_A, w_B, and $b_{A,B}$) used to compare 50 of Edgar Degas' (A) artworks and 50 of Gustave Dor's (B) artworks.

7. Compute the two partial test statistics $\lambda_h = \sum_{n=1}^{N} \mathbf{I}(T^*_{hn} \geq T_h)/N$ for $h = I, S$. Then the global test statistic is $\lambda(F, G) = \max(1 - \lambda_I, 1 - \lambda_S)$.

To test for the equivalence of two corpora, we compute the distance matrix M using *NCD*, then we compute (8). As a result, if both $\lambda(w_A, b_{A,B})$ and $\lambda(w_B, b_{A,B})$ are significant, then we consider the two corpora equivalent.

$$p_{\text{eqv}} = \max(\lambda(w_A, b_{A,B}), \lambda(w_B, b_{A,B})) \tag{8}$$

Experiment

Methodology

To evaluate the proposed statistical tests, we use datasets from two different domains; the classical archives MIDI dataset, which consists of 14,724 compositions by 843 distinct composers, and the Wikiart dataset, which consists of 19,052 paintings by 23 artists. There are two conditions, one where both corpora (A, B) have the same class (they are created by the same composer or artist), and another where the corpora have a different class. Therefore, the ground truth is calculated using (9), and the condition predicted by each statistical test is calculated using (10), with the standard significance level ($\alpha = 0.05$). To create corpora of different sizes, we randomly select artifacts without replacement belonging to the same class.

$$g(a, b) = \begin{cases} 0, & \text{if class}(a) \neq \text{class}(b) \\ 1, & \text{else} \end{cases} \tag{9}$$

$$\hat{g}(a, b) = \begin{cases} 0, & \text{if } p_{\text{eqv}} \geq \alpha \text{ or } p_{\text{diff}} < \alpha \\ 1, & \text{if } p_{\text{eqv}} < \alpha \text{ or } p_{\text{diff}} \geq \alpha \end{cases} \tag{10}$$

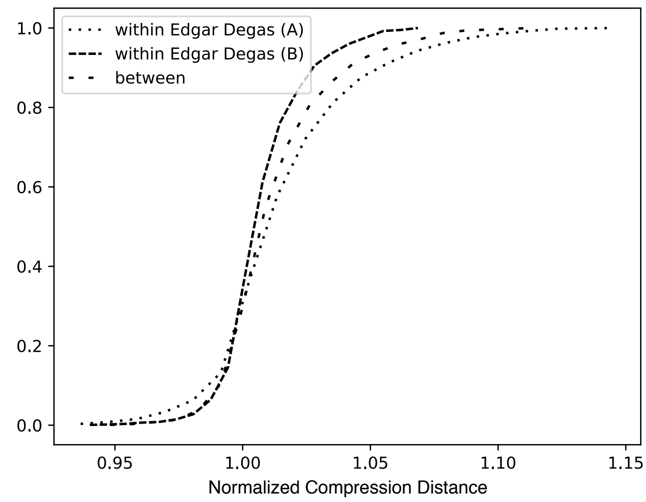


Figure 2: The cumulative *NCD* distributions (w_A, w_B, and $b_{A,B}$) used to compare two disjoint subsets of Edgar Degas' artwork, both of size 50.

Preprocessing Step	0	1	2
Wikiart	19052	19052	18874
Classical MIDI Archives	14724	12117	11943

Table 1: The corpus size after each preprocessing step

Data Pre-Processing

Since our test statistic takes the pairwise distance of all items within a corpora into consideration, having a number of duplicate items would artificially decrease values in w_A and w_B. As a result, we took the following steps to remove duplicate items in each dataset.

1. Remove all artifacts which belong to the same class and have the same title.
2. Remove all artifacts which belong to the same class and have a similarity greater than a threshold (t_{sim}).

We measure the similarity between two images using the structural similarity index (Wang et al. 2004), which takes structural information into account, rather than quantifying visible differences. To measure the similarity of two MIDI files, we extract a list of the pitches in the MIDI file ordered by onset time. Given compute time constraints, we only take the first 1000 notes into consideration. The following equation is used to quantify similarity, where $E(a, b)$ denotes the edit distance between two pitch sequences.

$$s = 1 - \frac{E(a, b)}{1000} \tag{11}$$

We set the similarity threshold (t_{sim}) at 0.75. Although this is quite conservative, we found that this did not eliminate too many artifacts, while providing confidence that duplicate artifacts are not included in the dataset. Table 1 lists the size of each corpus after each preprocessing step.

Data Representation

In order to avoid taking metadata, such as the title, composer, and author into consideration when computing the *NCD*, we do not use a binary representation of the MIDI files. Instead we create a representation which excludes irrelevant data. Since the velocity of MIDI note onsets is primarily based on the performer's interpretation of the composition, and in some cases may be set to a constant value if the MIDI file was created in a notation editor, we ignore this information. As a result, we represent a MIDI file as a sequence of onsets, offsets and time deltas. We represent onsets on the range $[0, 127]$, offsets on the range $[128, 255]$, and time deltas on the range $[256-)$. This results in a sequence of integers, which is then converted to a binary string before measuring the *NCD*. The representation used for images is much simpler. Each image is resized to have the shape 64×64, with three color channels (RGB), where each pixel is represented as an integer on the range $[0, 255]$.

Results

In Table 2 we present the results of 1000 trials, half of which have a ground truth of 0, and half which have a ground truth of 1, for a variety of corpora sizes. The *accuracy* (ACC), *true positive rate* (TPR), *false positive rate* (FPR), *true negative rate* (TNR), and *false negative rate* (FNR), are reported, using the formulas shown below, where n is the number of trials. *True positive* indicates trials in which the statistical test predicts 1 and the ground truth is also 1 ($\hat{g}(a,b) = 1 \wedge g(a,b) = 1$). Similarly, *true negative* indicates trials in which the statistical test predicts 0 and the ground truth is also 0 ($\hat{g}(a,b) = 0 \wedge g(a,b) = 0$). $\varepsilon = \varepsilon_I = \varepsilon_S$ denotes the equivalence range, which is normalized with respect to the length of $F = [f_i, i = 1, ..., n]$ and $G = [g_i, i = 1, ..., m]$ in Table 2. For example, if $\varepsilon = 0.1$ denotes an equivalence range of $(m+n) * 0.1$.

$$ACC = \frac{\sum \text{True positive} + \sum \text{True negative}}{n} \tag{12}$$

$$TPR = \frac{2 \sum \text{True positive}}{n} \tag{13}$$

$$TNR = \frac{2 \sum \text{True negative}}{n} \tag{14}$$

$$PPV = \frac{\sum \text{True positive}}{\sum \text{Predicted positive}} \tag{15}$$

$$NPV = \frac{\sum \text{True positive}}{\sum \text{Predicted negative}} \tag{16}$$

A robust statistical test, will minimize the probability of type I error (α), incorrectly rejecting a true null hypothesis, and type II error (β), incorrectly rejecting a true alternative hypothesis. The power of a statistical test is $1 - \beta$, which is equivalent to the *TNR* with respect to the test for difference (p_{diff}), and the *TPR* with respect to the test for equivalence (p_{eqv}). Since we also must verify that the tests minimize type I error, we provide the *TPR* and *TNR* which are equivalent to statistical sensitivity, for p_{diff} and p_{eqv} respectively. For each trial, we perform 1000 permutations, as this is what Marozzi suggests when estimating the power of a permutation test (2004).

Discussion

Given the degree of intra-corpus variation, and inter-corpus similarity, it is difficult to establish a ground truth for corpus comparison. In many cases, an artist or composer may explore several different sub-styles over the span of their career. Furthermore, artists and composers are often inspired by their colleagues, creating works that exhibit a greater than average degree of similarity. As a result, it would be unreasonable to expect extremely high values of accuracy. Nevertheless, according to Cohen, 0.80 is an adequate level for statistical power (1988), which most of the tests surpass. Overall, the results of the experiment demonstrate that the proposed tests provide a robust measurement of the stylistic difference between two corpora.

We used different values for ε to account for the decrease in variability of w_A, w_B and $b_{A,B}$ as the size of the corpora increases. For example, if two paintings are randomly selected from the work of a single artist, in some cases, given the variability of that artist's work, the mutual information between these two paintings will be fairly low. In other cases, when both paintings are part of the same sub-style, the mutual information may be fairly high. However, as we increase the number of paintings selected, stylistic tendencies will start to emerge, and the amount of mutual information amongst the selected paintings will converge. As a result, w_A and w_B will decrease in variability as the size of the corpora is increased, which allows us to decrease the size of the equivalence interval by decreasing ε.

The results in Table 2 show two trends. On average the statistical tests performed better on MIDI than on images. There are two possible explanations for this; composers may have a more consistent style than artists, or the representation we used for images is not optimized for comparison. However, the fact that images were not preprocessed, as we simply resized each image and extracted the raw pixel values, demonstrates that *NCD* is capable of finding commonalities in the raw data. Secondly, the statistical tests perform better on larger corpora than smaller corpora, which is primarily the result of decreasing stylistic variability as the size increases.

Since the strings that are being compared were quite long, the *NCD* between two items was heavily skewed towards 1, as shown in Figure 1 and 2. Consequently, we do not suggest interpreting these values as interval, but rather as ordinal values. Despite the skew of these values, discrepancies between w_A, w_B, and $b_{A,B}$ can be quite pronounced.

Application

There are several ways in which the proposed tests could be used. In the most basic sense, the tests could be used to compare the source corpus (C) with a corpus of artifacts generated by the SI system (G). The magnitude of p_{eqv} can indicate how similar the two corpora are. In the case that $p_{\text{eqv}} >= \alpha$, the test for difference can be used to determine if there is a significant difference between the two corpora. In addition, it may be of particular interest to measure the similarity of $\hat{C}_s$ and $\hat{G}_s$, which denotes the projection of C and G into a lower dimensional feature space s. For example, in the music domain, one could use a representation that only contains rhythmic information, and another that

	Test	Corpus A		Corpus B		ACC	TPR	TNR	PPV	NPV	ε
		size	classes	size	classes						
WikiArt	p_{diff}	25	23	25	23	0.85	0.96	0.75	0.79	0.95	-
	p_{eqv}	25	23	25	23	0.78	0.78	0.77	0.78	0.77	0.15
	p_{diff}	50	23	50	23	0.92	0.97	0.87	0.88	0.96	-
	p_{eqv}	50	23	50	23	0.86	0.85	0.87	0.87	0.85	0.1
	p_{diff}	100	23	100	23	0.94	0.98	0.90	0.90	0.98	-
	p_{eqv}	100	23	100	23	0.92	0.94	0.90	0.90	0.93	0.075
	p_{diff}	50	23	100	23	0.82	0.98	0.67	0.75	0.97	-
	p_{eqv}	50	23	100	23	0.88	0.87	0.88	0.88	0.88	0.0875
Classical Archvies	p_{diff}	25	74	25	74	0.98	0.99	0.97	0.97	0.99	-
	p_{eqv}	25	74	25	74	0.92	0.88	0.95	0.94	0.89	0.15
	p_{diff}	50	37	50	37	0.99	1.00	0.99	0.99	1.00	-
	p_{eqv}	50	37	50	37	0.91	0.92	0.90	0.90	0.92	0.1
	p_{diff}	100	20	100	20	0.99	1.00	0.99	0.99	1.00	-
	p_{eqv}	100	20	100	20	0.93	0.94	0.91	0.92	0.94	0.075
	p_{diff}	50	37	100	20	0.85	1.00	0.75	0.77	0.99	-
	p_{eqv}	50	37	100	20	0.89	0.87	0.91	0.91	0.87	0.0875

Table 2: The results of 1000 randomized trials for each statistical test $(p_{\text{eqv}}, p_{\text{diff}})$ using a variety of corpora sizes.

only contains information about the harmonic progression, to gauge the degree to which the SI emulates the rhythm, and harmonic progressions which characterize C.

These tests could also be used to assess the CAN (Elgammal et al. 2017), which attempts to produce visual art in a style that is distinct from those it is trained on. In this scenario, we would have a set of corpora on which the CAN is trained $(S = C_i : i = 1, ..., n)$, and for each $C_i \in S$ we would need to verify that $p_{\text{diff}} < \alpha$, using corrections for multiple hypothesis testing. Most importantly, since *NCD* operates on binary strings, these statistical tests are domain independent, as any digital data can be represented as a binary string.

Conclusion

Scientific progress is hindered in the absence of robust evaluation methodologies. This is an issue of particular contention in the field of computational creativity, as the subjective nature of assessments on creative artifacts can be problematic. In addition to issues of adequate domain knowledge, bias, and inter-rater reliability, the finite capacity of human participants limits the scalability of many evaluation approaches. This is a particular issue for SI systems, where the source corpus is often large, and the generated corpus is infinite. To address this issue, we propose CAEMSI for the evaluation of SI systems, providing compelling evidence that the statistical tests are reliable in two distinct domains. Future work involves further experimention with datasets from other domains, and the evaluation of generative systems with CAEMSI.

Acknowledgments

Thank you to Pierre R. Schwob for providing the Classical Archives MIDI dataset.

References

Amabile, T. M. 1982. Social psychology of creativity: A consensual assessment technique. *Journal of Personality and Social Psychology* 43(5):997–1013.

Ariza, C. 2009. The Interrogator as Critic : The Turing Test and the Evaluation of Generative Music Systems. *Computer Music Journal* 33(2):48–70.

Axelsson, S. 2010. Using normalized compression distance for classifying file fragments. In *Proceedings of the 5th International Conference on Availability, Reliability, and Security*, 641–646.

Bardera, A.; Feixas, M.; Boada, I.; and Sbert, M. 2010. Image registration by compression. *Information Sciences* 180(7):1121–1133.

Briot, J.-P., and Pachet, F. 2017. Music Generation by Deep Learning - Challenges and Directions. *arXiv preprint arXiv:1712.04371.*

Burns, K. 2015. Computing the creativeness of amusing advertisements: A Bayesian model of Burma-Shave's muse. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing (AIEDAM)* 29(1):109–128.

Cilibrasi, R., and Vitányi, P. M. B. 2005. Clustering by compression. *IEEE Transactions on Information Theory* 51(4):1523–1545.

Cilibrasi, R.; Vitányi, P.; and De Wolf, R. 2004. Algorithmic clustering of music. *Computer Music Journal* 28(4):49–67.

Cohen, J. 1988. *Statistical Power Analysis for the Behavioral Sciences.* New York: Lawrence Erlbaum Associates.

Dahlig, E., and Schaffrath, H. 1997. Judgements of human and machine authorship in real and artificial folksongs. *Computing in Musicology* 11:211–219.

Eerola, T.; Himberg, T.; Toiviainen, P.; and Louhivuori, J. 2006. Perceived complexity of western and African folk melodies by western and African listeners. *Psychology of Music* 34(3):337–371.

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 *Proceedings of the 8th International Conference on Computational Creativity (ICCC)*, 96–103.

Friedman, R. S., and Taylor, C. L. 2014. Exploring emotional responses to computationally created music. *Psychology of Aesthetics, Creativity, and the Arts* 8(1):87–95.

Gatt, A., and Krahmer, E. 2017. Survey of the State of the Art in Natural Language Generation: Core tasks, applications and evaluation. *Journal of Artificial Intelligence Research (JAIR)* 61(1):65–170.

Gonzalez Thomas, N.; Pasquier, P.; Eigenfeldt, A.; and Maxwell, J. 2013. Meta-Melo: A system and methodology for the comparison of melodic generation models. In *Proceedings of the 14th International Symposium on Music Information Retrieval (ISMIR)*, 561–566.

Kaufman, J. C.; Baer, J.; and Cole, J. C. 2009. Expertise, domains, and the consensual assessment technique. *Journal of Creative Behavior* 43(4):223–233.

Kocsor, A.; Kertész-Farkas, A.; Kaján, L.; and Pongor, S. 2006. Application of compression-based distance measures to protein sequence classification: a methodological study. *Bioinformatics* 22(4):407–412.

Lamb, C.; Brown, D. G.; and Clarke, C. L. A. 2015. Human Competence in Creativity Evaluation. In *Proceedings of the 6th International Conference on Computational Creativity*, 102–109.

Li, M., and Sleep, R. 2005. Genre Classification via an LZ78-Based String Kernel. In *Proceedings of the 6th International Symposium on Music Information Retrieval*, 252–259.

Li, M.; Chen, X.; Li, X.; Ma, B.; and Vitányi, P. M. B. 2004. The Similarity Metric. *IEEE Transactions on Information Theory* 50(12):3250–3264.

Liang, F.; Gotham, M.; Johnson, M.; and Shotton, J. 2017. Automatic Stylistic Composition of Bach Chorales with Deep LSTM. In *Proceedings of the 18th International Society for Music Information Retrieval Conference*, 449–456.

Maher, M., and Fisher, D. 2012. Using AI to evaluate creative designs. In *Proceedings of the 2nd International Conference on Design Creativity*, 45–54.

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.

Marozzi, M. 2004. Some remarks about the number of permutations one should consider to perform a permutation test. *Statistica* 64(1):193–201.

Moffat, D. C., and Kelly, M. 2006. An investigation into people's bias against computational creativity in music composition. In *Proceedings of the 3rd International Joint Workshop on Computational Creativity*.

Norton, D.; Heath, D.; and Ventura, D. 2015. Accounting for Bias in the Evaluation of Creative Computational Systems: An Assessment of DARCI. In *Proceedings of the 6th International Conference on Computational Creativity*, 31–39.

Pasquier, P.; Burnett, A.; Thomas, N. G.; Technology, A.; Maxwell, J. B.; and Loughin, T. 2016. Investigating Listener Bias Against Musical Metacreativity Introduction : Computational Creativity. In *Proceedings of the 7th International Conference on Computational Creativity*, 47–56.

Pasquier, P.; Eigenfeldt, A.; Bown, O.; and Dubnov, S. 2017. An Introduction to Musical Metacreation. *Computers in Entertainment* 14(2):1–14.

Pearce, M. T., and Wiggins, G. a. 2007. Evaluating Cognitive Models of Musical Composition. In *Proceedings of the 4th International Joint Workshop on Computational creativity*, 73–80.

Pearce, M.; Meredith, D.; and Wiggins, G. 2002. Motivations and Methodologies for Automation of the Compositional Process. *Musicae Scientiae* 6(2):119–147.

Pearce, M. T. 2005. *The Construction and Evaluation of Statistical Models of Melodic Structure In Music Perception and Composition*. Ph.D. Dissertation, City, University of London.

Pesarin, F.; Salmaso, L.; Carrozzo, E.; and Arboretti, R. 2016. Unionintersection permutation solution for two-sample equivalence testing. *Statistics and Computing* 26(3):693–701.

Ritchie, G. 2007. Some empirical criteria for attributing creativity to a computer program. *Minds and Machines* 17(1):67–99.

Rocha, R.; Ribeiro, A.; and El, E. 2016. Regent-Dependent Creativity : A Domain Independent Metric for the Assessment of Creative Artifacts. In *Proceedings of the 7th International Conference on Computational Creativity*, 68–75.

Roy, S. 1953. On a heuristic method of test construction and its use in multivariate analysis. *The Annals of Mathematical Statistics* 24(2):220–238.

Schedl, M.; Flexer, A.; and Urbano, J. 2013. The neglected user in music information retrieval research. *Journal of Intelligent Information Systems* 41(3):523–539.

Simpson, S. L.; Lyday, R. G.; Hayasaka, S.; Marsh, A. P.; and Laurienti, P. J. 2013. A permutation testing framework to compare groups of brain networks. *Frontiers in Computational Neuroscience* 7(11):1–13.

Solomonoff, R. 1964. A formal theory of inductive inference. Part I. *Information and Control* 7(1):1–22.

Väyrynen, J., and Tapiovaara, T. 2010. Normalized compression distance as an automatic MT evaluation metric. In *Proceedings of the Joint 5th Workshop on Statistical Machine Translation and MetricsMATR*, 343–348.

Wang, Z.; Bovik, A. C.; Sheikh, H. R.; and Simoncelli, E. P. 2004. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing* 13(4):600–612.

Yang, L.-C.; Chou, S.-Y.; and Yang, Y.-H. 2017. MidiNet: A Convolutional Generative Adversarial Network for Symbolic-domain Music Generation. In *Proceedings of the 18th International Society for Music Information Retrieval Conference*, 324–331.

Computational Creativity for Valid Rube Goldberg Machines

Xiou Ge
University of Illinois at Urbana-Champaign
xiouge2@illinois.edu

Jinjun Xiong
IBM T.J. Watson Research Center
jinjun@us.ibm.com

Lav R. Varshney
University of Illinois at Urbana-Champaign
varshney@illinois.edu

Abstract

Laboratory activity is an indispensable part of science and engineering education. To develop children's interest in science and engineering, we want to create hands-on activities using artificial intelligence. In this paper, we first describe the use of case-based reasoning (CBR) and an existing knowledge base to yield a combinatorial design space for experiments. We then apply automated planning techniques to generate experiment procedures. We further use functional modeling to represent the experiment devices and demonstrate how that representation enables the planner to generate a valid Rube Goldberg Machine. Finally, a semantic similarity metric is proposed to evaluate the quality of a generated chain of experiments.

Introduction

In Science Olympiad[1] competitions, middle school and high school students from all over the country participate in science experiment design contests to demonstrate relevant scientific concepts. That there are competitions already shows that creating science experiments is not easy. Designing experiments requires not only immense knowledge about the domain but also sufficient information about the properties of available materials. More importantly, students also need imagination and organization skills to arrange the materials rationally and plan out the details of data collection.

Consider building an artificial intelligence system to create novel science experiments. With scientific knowledge and sample experiments in hand, forming useful representations of this data is the key challenge. Much past work has attempted to design experiments for scientific research itself rather than for students. Early work can be traced back to MOLGEN (Stefik, 1981), a knowledge-based system that plans molecular genetics experiments using hierarchical planning techniques. A layered control structure was also introduced to enable meta-planning. MOLGEN focused on the detailed domain knowledge and required much human intervention for a valid experiment plan to be generated. Such systems are not suitable for generating engaging science experiments for students.

Beyond single experiments, it may be more engaging for students to connect a series of devices to form a chain. There is, in fact, a Rube Goldberg Machine (RGM) competition in Science Olympiad called *Mission Possible*[2] for creating chain-reaction machines. The Rube Goldberg Inc also organizes a contest[3] specifically for designing RGM. RGM design has also been brought into class to help teaching. Sharpe, Qin, and Recktenwald (2015) have shown that an RGM-like device setup is good at engaging students and helping them gain deeper understanding of difficult concepts. In fact, Wu et al. (2015) have started to build valid RGMs from the perspective of scene understanding using deep learning and a simulation engine.

In creating such "comically-involved, complicated inventions laboriously contrived to perform a simple operation", judging criteria explicitly require a notion of surprise. As a recent rule book says, "RGMs should work but they also need to capture attention. The more theatrical and funny your machine is, the better it will score!".

In order to build a system that generates creative RGM ideas, we answer several key questions.

- How can knowledge about experimental materials be represented to enable similarity-based retrieval?
- Which class of parts in the existing knowledge base can be used for material substitution?
- How can chains of experiments be generated?
- How can procedure instructions to build RGMs be generated automatically?
- Which generated chain is the most interesting and has highest educational value?

We build algorithmic components to address these questions; putting them together yields a full computational creativity system to generate valid RGMs and assess their quality. By *creativity*, we mean simultaneously achieving novelty and domain-specific quality.

Fig. 1 shows the basic structure of our system. First, we propose a feature-based case representation for experiment materials and adapt mixed-attribute dissimilarity measures from data mining into a distance metric for material retrieval. We also suggest using WordNet to generate more possible substitution materials with the help of word sense

[1]https://www.soinc.org/

[2]https://www.soinc.org/mission-possible-c

[3]https://www.rubegoldberg.com/education/contest/

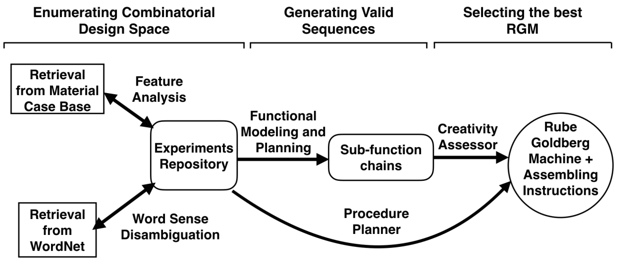


Figure 1: System Structure

disambiguation. Inspired by engineering design, we apply the functional modeling language to represent units used in constructing RGMs and use a forward planner to generate chains of experiments. Procedure plans for building RGMs are also suggested by a partial order planner. We generate examples of experiment chains using our system and propose a creativity evaluation metric for RGMs based on rules from the student competitions and semantic similarity computation using word vectors.

Choosing Materials

Designing science experiments and projects is similar to culinary recipe creation in that both involve suggesting sets of materials and procedures. An AI system with the capability of suggesting unusual combination of materials for a known goal often amuses people (Olteţeanu and Falomir, 2016) and is considered creative according to Bayesian surprise (Itti and Baldi, 2006; Varshney et al., 2013a; França et al., 2016).

In RGM generation, however, not all combinations of devices can be sequenced into a chain due to common material constraints in consecutive devices. Viewing an experimental device as a decomposable system made up of experimental materials, new experiments can be designed if one has access to a set of materials similar to ones in inspiration experiments. By doing this, the constrained combinatorial design space of materials for generating valid RGMs can be enlarged considerably.

Morris et al. (2012); Pinel, Varshney, and Bhattacharjya (2015) suggest that culinary ingredients may be classified into a hierarchy of categories. To generate a recipe, a certain number of ingredients are selected from each category based on pairing rules learned from existing recipes. Unlike in culinary creativity where a single taxonomy of ingredients is applicable to most recipe generation tasks, a suitable classification for one case will likely fail in other cases for experiment generation since the usage of materials is context-dependent.

This issue is more apparent when we try to design experiments using materials that are commonly found at home since a single material may serve different purposes in different scenarios. For example, one might logically classify a marble ball and steel ball into the same category due to their common shape. This would work to roll different objects that perform rotational motion down a ramp and make a series of measurements and observations. However in the Gauss rifle experiment, the marble is not a good substitute for the steel ball since the marble is not ferromagnetic. The marble will not be attracted and accelerate towards the magnet to produce enough momentum to eject the steel bullet.

In addition, a single classification will restrict creativity by dismissing many possible candidates for material substitution. For example, keyboard is put under the computer accessories category whereas wood plank is classified as a type of construction material. In such a taxonomy, keyboard is very distant from wood plank. However, if features such as shape (both approximately cuboid) and surface finish (both have at least one flat surface) are provided, the keyboard will be considered in the set of replacement materials for the wood plank. Therefore, specific feature descriptions of materials are more pragmatic than a comprehensive and refined taxonomy of materials for experiment generation.

Feature-based retrieval

Since science experiment design is knowledge intensive, we want to take advantage of existing data through proper knowledge representation. To ensure the validity of experiment, we start by considering existing experiments as cases and experiment materials as the varying factor.

In engineering design, CBR methods have been applied for material selection (Ashby et al., 2004). Material attributes of mixed types are analyzed and stored in the case base. Based on requirements specified by the designer, a list of materials can be retrieved from the case base. Experiment material substitution is similar to material selection in that features of the original material can be used as key terms to search the knowledge base. Accurate feature information can be extracted from material vendors' websites and refined using crowdsourcing platforms (Demartini et al., 2017). Olteţeanu and Falomir (2016) have demonstrated the effectiveness of feature-based retrieval in creative replacement of everyday objects. Good candidates for replacements are those having high similarity with the original material. In our application, we use the nearest neighbor strategy to search for substitution material.

Material attributes include basic features such as length, shape, or weight, but also context-specific properties such as melting point or electrical conductivity. By referencing the material ontology defined in Ashino (2010), we built a material property ontology to standardize the use of feature names to enable the sharing of material information among different databases using the Protégé ontology editor (Musen, 2015).

Note that material features are not restricted to numeric attributes, but could also include nominal, binary, and ordinal attributes. Han and Kamber (2000) introduced dissimilarity measures for attributes of mixed types. We define the distance metric for our nearest neighbor retrieval in the same manner. Numeric, nominal, binary, and ordinal attributes are dealt with differently as follows. In all equations, x_{if} is the value of attribute f for object i.

- For numeric attributes, the distance is normalized with the difference between the highest and lowest value possible for the particular attribute:

$$d_{ij}^{(f)} = \frac{|x_{if} - x_{jf}|}{\max_h x_{hf} - \min_h x_{hf}} \tag{1}$$

- For nominal or binary attributes:

$$d_{ij}^{(f)} = \begin{cases} 0, & \text{if } x_{if} = x_{jf}. \\ 1, & \text{otherwise.} \end{cases} \tag{2}$$

- For ordinal attributes, first count the number of possible ordered states M_f. Then convert the attribute to its corresponding rank, $r_{if} \in \{1, ..., M_f\}$. The rank is normalized and mapped to $[0, 1]$ by the following:

$$Z_{if} = \frac{r_{if} - 1}{M_f - 1} \tag{3}$$

  After conversion, values for ordinal attributes are treated the same way as numerical attributes to compute $d_{ij}^{(f)}$.

- Since not all material features are relevant to a particular experiment, domain experts could label the essential material features to the set E and the less important features to set L. We assign higher weights to more relevant features and lower weights to the less relevant ones when computing the overall distance between material pairs to ensure the replaceability of the retrieved material. The overall distance $d(i, j)$ between experiment material i and j is defined as:

$$d(i,j) = w\frac{\sum_{f \in E} \delta_{ij}^{(f)} d_{ij}^{(f)}}{\sum_{f \in E} \delta_{ij}^{(f)}} + (1-w)\frac{\sum_{f \in L} \delta_{ij}^{(f)} d_{ij}^{(f)}}{\sum_{f \in L} \delta_{ij}^{(f)}} \tag{4}$$

  where $\delta_{ij}^{(f)} \in \{0, 1\}$ indicates whether attribute f appears in both material i and j. $\delta_{ij}^{(f)} = 0$ if an attribute is missing in either material i or j; $\delta_{ij}^{(f)} = 1$ otherwise.

An example of the described knowledge representation and retrieved substitution material is shown in Fig. 2. Material features essential to the problem scenario are highlighted. Constraints are used to check the compatibility of materials within an experiment. The generated combination will be dismissed if materials in a single experiment do not satisfy the constraints specified. Constraints are also used for RGM generation discussed later in this paper where compatibility between different components is essential.

Retrieval from general semantic resources

We propose to augment material substitution retrieval using WordNet (Miller, 1995), a general-purpose knowledge base. In WordNet, nouns are organized into a hierarchical structure in which words are linked by "is a" relationships in terms of their meanings. A more generic concept is referred to as a hypernym whereas a specific instance of a concept is referred to as a hyponym. A hyponym inherits all features of the more generic concept and adds features that distinguishes it from superordinate and sister terms (Touretzky, 1986). Although features of entities are not explicitly specified for each synset entry in WordNet, one can still search for

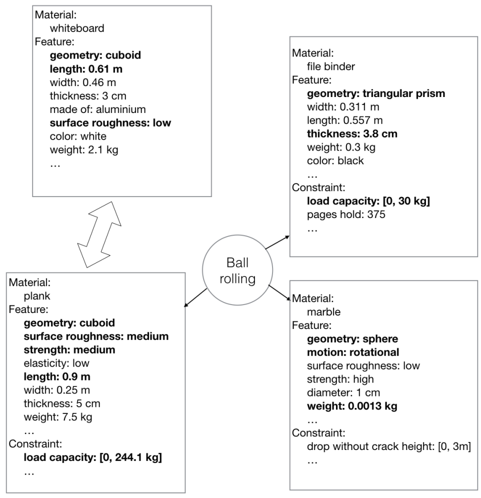


Figure 2: Replacement found by nearest neighbor retrieval

entities with similar features by traversing through the hierarchy. One way of searching is by first looking up the hypernym of the target word and then listing out all hyponyms of the hypernym.

Fig. 3 shows the hierarchical structure in WordNet and selected terms returned on a possible query. In a scenario of building a ramp, terms returned like sheet metal, panel, and plate glass are all good replacements for a board. Also, a query of the hyponym of the target word can also return good candidates like surfboard, ironing board, and wallboard. This method augments the substitution set without requiring extensive human effort in labeling features for experiment materials.

A problem one might face is the material term might be a polysemous word. In WordNet, words are grouped into sets of synonyms called synsets. An example of synset encoding is 'board.n.01', in which the first entry is the word itself, second entry is the part of speech (POS) tag and the third entry represents the index of sense that the term corresponds to. When looking up a word, all possible synsets associated with different meanings of the word will be returned. To search for substitution materials for experiments, the exact synset entry that the original material corresponds to is required. However, the synset entry will not be available unless someone assigns the label manually or using Word Sense Disambiguation (WSD) techniques.

For our application, we use a Support Vector Machine (SVM) classifier to disambiguate the sense of a target word. The training data for the classifier is a list of example sentences that include the target words tagged with corresponding sense labels. We use word embeddings to represent the contextual features since they are more efficient for training and better at capturing relationships among concepts. After the classifier has been trained, it can predict sense labels for previously unseen examples based on the likelihood of each sense given the contextual features (Zhong and Ng, 2010).

As an example, we trained a linear SVM to disambiguate

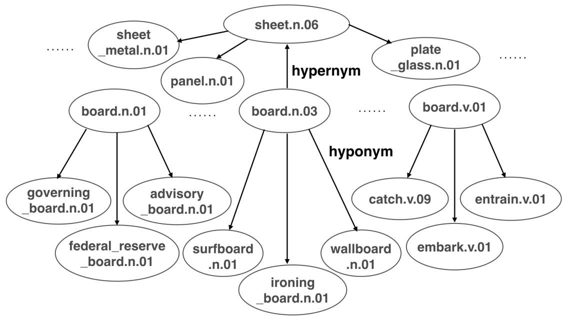


Figure 3: WordNet Hierarchy

the three most common senses of the word “board”. We collected 176 examples in total from several resources[4,5,6] to form a balanced dataset for training. We cleaned up the context corpora by removing punctuations, non-alphabetic characters, and common stop words. Remaining words are converted to their lemma forms in lower case.

A window size of five words on each sides of the target word is used to represent the context. Words within the window are mapped to a list of embeddings $W = \{\mathbf{v}_{-5}, ..., \mathbf{v}_{-1}, \mathbf{v}_1, ..., \mathbf{v}_5\}$. The word embeddings we use are obtained by training the skip-gram word2vec model (Mikolov et al., 2013) available in gensim package (Řehůřek and Sojka, 2010) with Wikicorpus scraped from the science domain. Similar to Iacobacci, Pilehvar, and Navigli (2016), we use the average strategy by computing the centroid of embeddings of the selected surrounding words to obtain the context vector.

$$C = \frac{\sum_{i \in W} \mathbf{v}_i}{|W|} \tag{5}$$

After extracting features for all examples, the set of contextual features and sense label pairs $\{(C_1, S_1), (C_2, S_2), ..., (C_n, S_n)\}$ are used to train the linear SVM. To test the performance of the SVM classifier, we run 5-fold cross-validation on the entire dataset and the accuracy is $M = 0.77, SD = 0.07$.

Rube Goldberg Machine Generation

Experiential learning activities are not limited to the conventional controlled experiment setting where repeated measurements are done to verify certain physical laws or relationships. Instead, learning concepts by building an RGM may be more engaging for students. In an RGM, a series of devices are setup in a way such that one device triggers another in a sequence. Along the chain reactions, many different science and engineering concepts are demonstrated. Learning could be more entertaining if the advisor could suggest possible ideas of building an RGM.

In science projects for students, an experimental setup typically has some function. For instance, a ramp can be considered as a module that performs kinetic energy and potential energy conversion for the object rolling on it. These modules are also frequently used in building an RGM. Given the many possibilities of modules made possible due to combination of materials, the design space will be even bigger if we can build a chain of these modules.

[4]http://www.comp.nus.edu.sg/ nlp/corpora.html

[5]http://sentence.yourdictionary.com/Board

[6]http://www.manythings.org/sentences/words/board/1.html

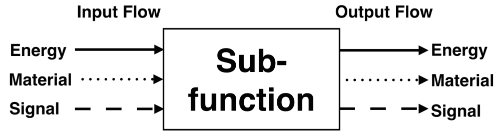


Figure 4: Functional modeling representation of devices

sub-function:	ConvertGPEtoKE
precondition:	Energy(Human)
effect:	Material(Wheel), Energy(K.E.), ¬ Energy(Human)

Table 1: Sub-function Schema

Functional Modeling Representation

Before thinking about automatic chain synthesis, it is important to come up with a systematic way to describe and represent the devices and relationship between them. Given the diversity of devices and the multi-disciplinary knowledge involved, coming up with a consistent knowledge representation is non-trivial. In engineering design, a holistic design is usually disassembled into sub-modules for conceptual analysis. Pahl, Beitz, and Wallace (1984) represent functional modules using block diagrams and call them sub-functions. Each sub-function block has input and output flows that fall in three main categories: energy, material, and signals. Each sub-function can be mapped to a corresponding physical embodiment. As suggested by Bohm, Stone, and Szykman (2005), the functional model allows multiple different types of input and output flows for each block to ensure completeness in knowledge representation. Real mapping examples such as power screwdriver and automobile seat can be found in Hirtz et al. (2002).

Devices in RGMs can be represented using sub-functions. We find the taxonomy of modeling vocabulary defined by Hirtz et al. (2002) useful for representing the devices in an RGM. By referencing the modeling language, we formally analyze the function of each device and its input and output to obtain sub-function representations of device units as shown in Fig. 4. For example, a Gauss rifle device can be interpreted as a system that converts magnetic potential energy to kinetic energy (K.E.). Human effort in the *Energy* category is the input to trigger the system. Both a steel ball in the *Material* category and K.E. in the *Energy* are outputs of the system.

For our application, we represent each sub-function block as planning operators using a STRIPS-like representation. Input and output flows of each sub-function are represented as preconditions and effects associated with the operator re-

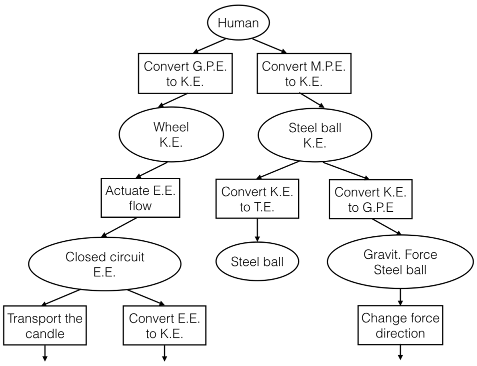


Figure 5: Part of a state space graph

spectively. An example of such a sub-function operator is shown in Table 1. A planning problem can be formed when a set of sub-function operators, initial input, and goal output are specified. We use a forward search algorithm for planning. Part of a possible state space graph expanded by the forward planner is shown in Fig. 5. In the state space graph, each oval represents a world state described by the flows. At first, only the initial state appears in the state space graph. An applicable operator (represented by the rectangles) can be added after a state if its preconditions are supported in that state. A new state is also added to the graph due to the change brought about by the operator. The algorithm terminates when the goal condition is found in a new state. A valid chain of functional blocks is a path in the graph from initial condition to the goal. By mapping each functional block in the chain to its corresponding physical embodiment, we get an RGM. Since materials used in each device are also binded with constraints, these constraints can be used to check the compatibility between adjacent units for physical embodiment selection. Fig. 6 shows several generated sub-function chains and corresponding RGMs.

The idea of building RGMs from science project components based on their input/output matching and compatibility can be further extended to the design of actual engineering systems and products (Li et al., 2013). Knowledge representation in our system is distinct from other planning applications like the story generation (Riedl and Young, 2006). Functional modeling language better reveals the scientific concepts behind the engineering processes and is thus better for educating students.

Suggesting Assembling Procedure Plans

Procedural instructions for building each module and connecting different modules into a chain are equally important. Much research has been done to create procedural artifacts including business processes (Heinrich, Klier, and Zimmermann, 2015), manufacturing simulations, and space missions. In computational creativity, efforts have also been made to create procedural artifacts. In Chef Watson, a graph matching and merging approach has been proposed to create recipe steps (Pinel, Varshney, and Bhattacharjya, 2015). Existing recipe instructions are parsed into directed acyclic graphs in which nodes are ingredients and discrete actions. We find the planning approach appropriate for our system since not all actions in an experiment are associated with a material or a concrete entity.

action:	placeOn(A, B)
precondition:	have(A), have(B), canHandle(A), withinLoadCapacity(A, B)
effect:	on(A, B)

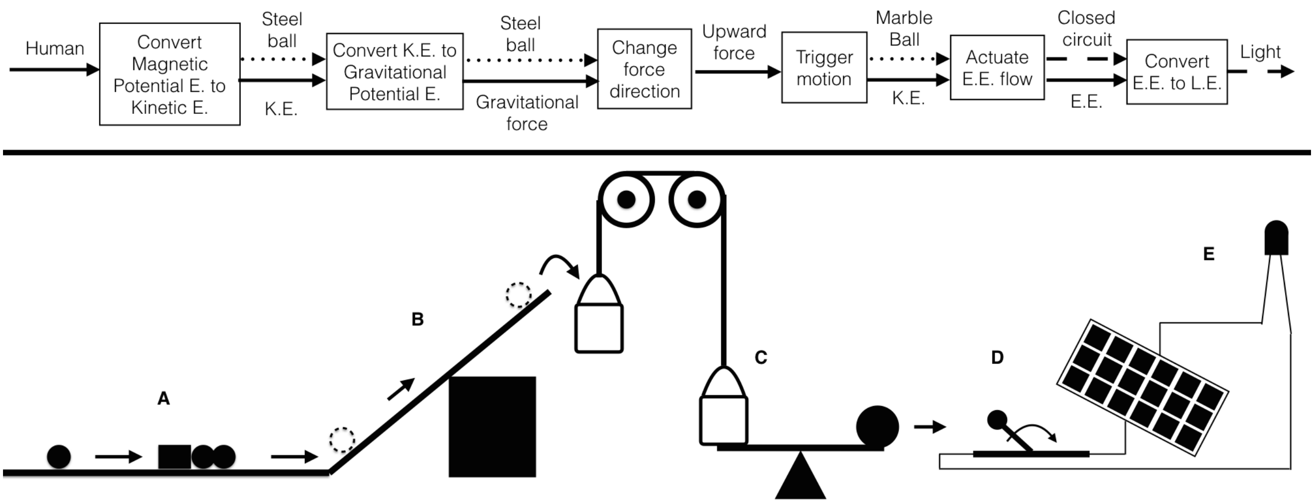
Table 2: Action Schema

For generating procedures, the initial world state can be described with as a set of literals such as available materials, constraints. The desired outcome can be stated as propositions to be satisfied for the goal state. Actions are represented using operators that include a set of preconditions and effects of executing the actions. For our problem, we use a partial order planner to generate plans of assembling procedure. At every iteration, the planner randomly selects an operator from the knowledge base that satisfy any goal conditions, referred to as open condition flaws. Once an action has been instantiated, the preconditions of this action becomes the new open condition flaws. On the next iteration, operators are selected to repair both old and new flaws. A causal link is constructed between action s_1 and s_2 via a specific condition e, represented as $s_1 \xrightarrow{e} s_2$, when execution of s_2 requires condition e established by s_1. The algorithm terminates when each precondition of each action is supported by the effects of a previous action or by conditions in the initial world state. A causal chain of actions that transform the initial world state to the goal state is thus a logical procedure for conducting the experiment. Since partial order planning enforces causal dependency, generated plans are ensured to be valid.

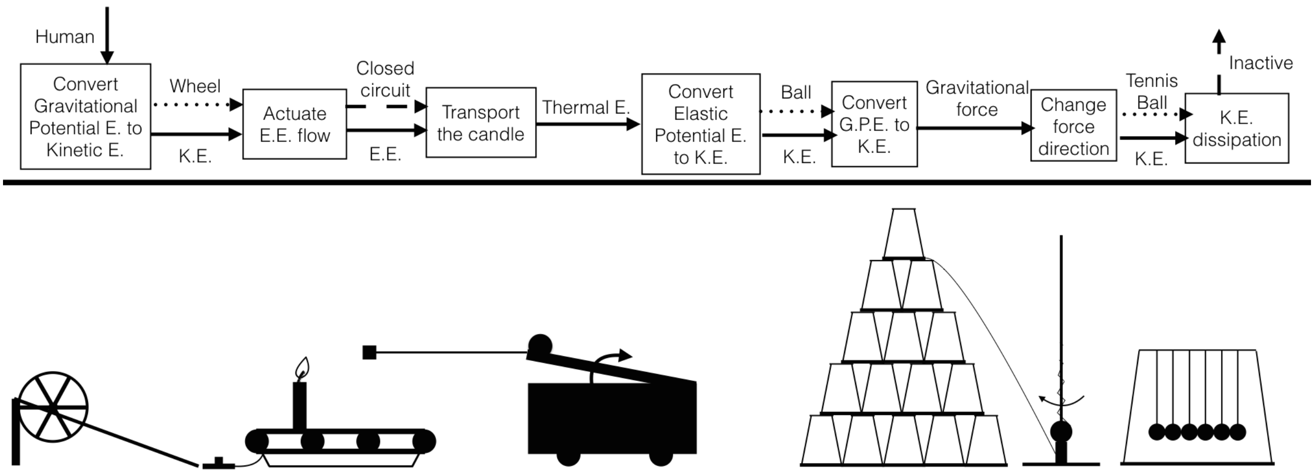
For our problem setting, the STRIPS-like representation is again used to define the planning problem. An example action schema is shown in Table 2. A plan is generated by the partial order planner for the Gauss rifle experiment see Fig. 7. Natural language generation techniques mentioned by Wasko and Dale (1999) can be applied to generate human readable texts from plans.

Selection through Creativity Evaluation

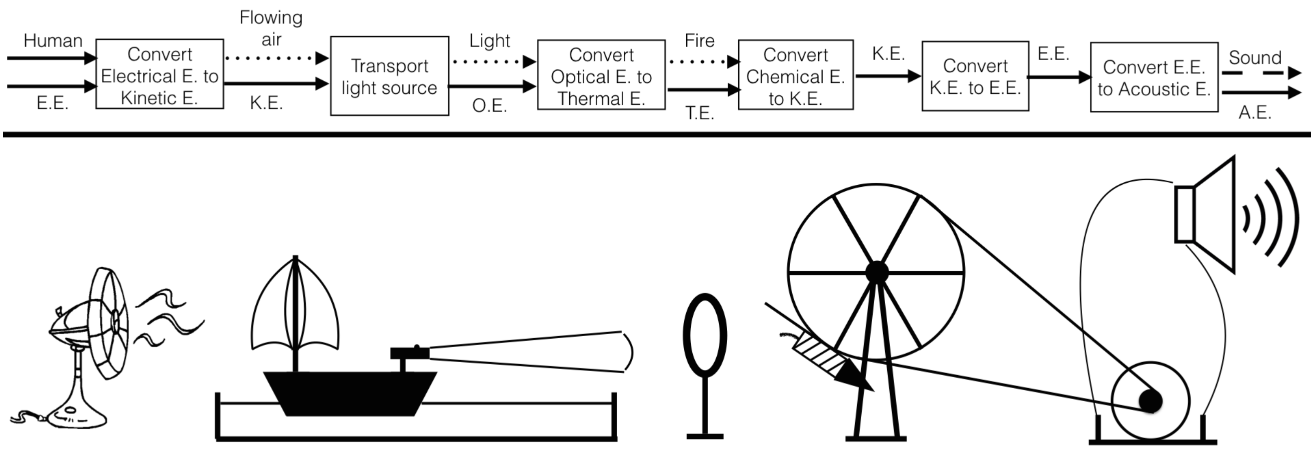
As part of the computational creativity system, internal assessment of creativity of the generated artifact is essential (Varshney et al., 2013b). We find that ramps, spirals, domino, and other physical contact-based devices are very common in RGMs. Creating a machine simply by repeating these components may not be as engaging as those that involve greater variety of reactions. Rules from the "Mission Possible" competition give higher scores to RGMs using components from different categories and having more energy transfers. Considering these rules, we count the number of energy transformations, disciplines, and concepts involved in the three generated RGM examples and display the result in Fig. 8. The example in Fig. 6c might outperform the other two since it involves more energy transfor-



(a) Example a



(b) Example b



(c) Example c

Figure 6: Generated chain of sub-functions and corresponding RGMs

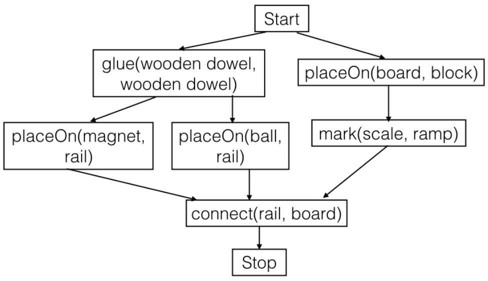


Figure 7: Assembling Plan

Transition	Similarity score
A→B	0.385
B→C	0.337
C→D	0.136
D→E	0.320

Table 3: Cosine score for concepts in each transition

mations and fields of knowledges with comparable number of concepts. We use the above metrics for creativity evaluation since they directly indicate the knowledge content of generated artifacts. According to Cohen (1999), knowledge is one of the three key requirements for creative behavior.

For educational purposes, we think that an engaging chain should demonstrate concepts from different disciplines. In particular, the more different the concepts involved in adjacent devices, the more novel the chain and thus should be given higher priority. We analyzed the concepts involved in each devices; concepts binded with each device in the generated example (a) is shown in Table 4. To measure the extent of transition in concept domains, we need to map each concept to the pre-trained vector representations. Cosine metric has been used to measure the semantic similarity of words in vector representations. We adopt the cosine similarity and compute the transition score by the following:

$$\text{Score} = \frac{(\sum_{i\in C_1} \mathbf{v}_i)^T(\sum_{i\in C_2} \mathbf{v}_j)}{\|\sum_{i\in C_1} \mathbf{v}_i\|\|\sum_{i\in C_2} \mathbf{v}_j\|} \quad (6)$$

where $\mathbf{v}$ is the distributed representation of a concept, C_1

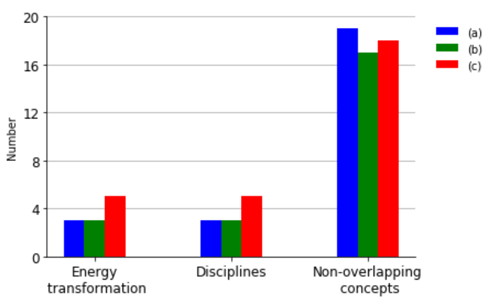


Figure 8: Quality measures

and C_2 are the sets of concepts related to two adjacent components respectively. The transition scores shown in Table 3 agree with our intuition as transitions across different disciplines have lower cosine score than those within a discipline. A chain with low cosine score and more cross-discipline transitions should be considered more creative.

Conclusion

We have described a full computational creativity system that generates RGMs. Several contributions of the system have been discussed. To recap, both CBR and lexical substitution techniques are demonstrated to suggest high quality replacement material. We also apply functional modeling concepts to device representation and generate chains of experiments using a forward planner. Classical planning concepts are applied to represent RGM construction problems and a partial order planner is used to generate procedural instructions. To guide creative artifact selection, we prioritize chains involving the most discipline transitions by computing semantic similarity of relevant concepts. We will continue to develop the system and expand the knowledge base by encoding more components into their corresponding subfunction representations and identifying the related concepts to those components via crowdsourcing.

As future work, we can measure the creativity of generated chains by analyzing the response of human audiences, e.g. through eye-tracking experiments, to understand devices in the chain that are attractors, sustainers, and relators (Candy and Bilda, 2009; Edmonds, Muller, and Connell, 2006). This could potentially help us evaluate the comicality of generated RGMs, which is difficult to measure using which is difficult to measure otherwise.

Acknowledgments

This work was funded in part by the IBM-Illinois Center for Cognitive Computing Systems Research (C3SR), a research collaboration as part of the IBM AI Horizons Network. We thank H. Gong and T. Sakakini for pointing us to the science domain wiki corpus used for the word2vec model.

References

Ashby, M.; Brchet, Y.; Cebon, D.; and Salvo, L. 2004. Selection strategies for materials and processes. *Mater. Des.* 25(1):51–67.

Ashino, T. 2010. Materials ontology: An infrastructure for exchanging materials information and knowledge. *Data Sci. J.* 9:54–61.

Bohm, M. R.; Stone, R. B.; and Szykman, S. 2005. Enhancing virtual product representations for advanced design repository systems. *J. Comput. Inf. Sci. Eng.* 5(2):360–372.

Candy, L., and Bilda, Z. 2009. Understanding and evaluating creativity. In *Proc. 7th ACM Conf. Creat. Cogn. (C&C '09)*, 497–498.

Cohen, H. 1999. Colouring without seeing: A problem in machine creativity. *Special issue on AISB99: Creativity in the arts and sciences* 102:26–35.

A	B	C	D	E
Momentum	Gravity	Torque	Semiconductor	Resistance
Kinetic energy	Conservation of energy	Balance of moments	Resistance	Current
Newton's 2nd Law	Free Fall	Gravity	Closed loop circuit	Voltage
Acceleration	Friction		Power	LED
Magnetism	Newton's 2nd Law			Luminance
Conservation of energy				

Table 4: Concepts involved in each device for example (a)

Demartini, G.; Difallah, D. E.; Gadiraju, U.; and Catasta, M. 2017. An introduction to hybrid human-machine information systems. *Found. Trends Web Sci.* 7(1):1–87.

Edmonds, E.; Muller, L.; and Connell, M. 2006. On creative engagement. *Vis. Commun.* 5(3):307–322.

França, C.; Goes, L. F. W.; Amorim, Á.; Rocha, R.; and Ribeiro da Silva, A. 2016. Regent-dependent creativity: A domain independent metric for the assessment of creative artifacts. In *Proc. Int. Conf. Comput. Creativity (ICCC 2016)*, 68–75.

Han, J., and Kamber, M. 2000. *Data Mining: Concepts and Techniques*. San Francisco, CA, USA: Morgan Kaufmann.

Heinrich, B.; Klier, M.; and Zimmermann, S. 2015. Automated planning of process models: Design of a novel approach to construct exclusive choices. *Decis. Support Syst.* 78:1–14.

Hirtz, J.; Stone, R. B.; McAdams, D. A.; Szykman, S.; and Wood, K. L. 2002. A functional basis for engineering design: Reconciling and evolving previous efforts. *Res. Eng. Des.* 13(2):65–82.

Iacobacci, I.; Pilehvar, M. T.; and Navigli, R. 2016. Embeddings for word sense disambiguation: An evaluation study. In *Proc. Assoc. Comput. Linguist. Annu. Meet. (ACL 2016)*, 897–907.

Itti, L., and Baldi, P. 2006. Bayesian surprise attracts human attention. In *Adv. Neural Info. Process. Syst. 18 (NIPS 2005)*. 547–554.

Li, X.; Zhang, Z.-N.; Liu, Z.-L.; and Xie, Y.-B. 2013. A novel semi-heuristic planning approach for automated conceptual design synthesis. *Proc. Inst. Mech. Eng., C J. Mech. Eng. Sci.* 227(10):2291–2305.

Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G. S.; and Dean, J. 2013. Distributed representations of words and phrases and their compositionality. In *Adv. Neural Info. Process. Syst. 26 (NIPS 2013)*. 3111–3119.

Miller, G. A. 1995. Wordnet: A lexical database for English. *Commun. ACM* 38(11):39–41.

Morris, R. G.; Burton, S. H.; Bodily, P. M.; and Ventura, D. 2012. Soup over beans of pure joy: Culinary ruminations of an artificial chef. In *Proc. Int. Conf. Comput. Creativity (ICCC 2012)*, 119–125.

Musen, M. A. 2015. The protégé project: A look back and a look forward. *ACM AI Matters* 1(4):4–12.

Olteţeanu, A.-M., and Falomir, Z. 2016. Object replacement and object composition in a creative cognitive system. Towards a computational solver of the alternative uses test. *Cogn. Syst. Res.* 39(C):15–32.

Pahl, G.; Beitz, W.; and Wallace, K. 1984. *Engineering Design*. Design Council.

Pinel, F.; Varshney, L. R.; and Bhattacharjya, D. 2015. A culinary computational creativity system. In *Computational Creativity Research: Towards Creative Machines*. Springer. 327–346.

Řehůřek, R., and Sojka, P. 2010. Software framework for topic modelling with large corpora. In *Proc. 10th Int. Conf. Lang. Resour. Eval. (LREC 2010)*, 45–50.

Riedl, M. O., and Young, R. M. 2006. Story planning as exploratory creativity: Techniques for expanding the narrative search space. *New Gener. Comput.* 24(3):303–323.

Sharpe, T.; Qin, G.; and Recktenwald, G. W. 2015. A compact device for inductive instruction in general physics. In *Proc. 2015 ASEE Annu. Conf. Expos.*

Stefik, M. 1981. Planning with constraints (MOLGEN: Part 1). *Artif. Intell.* 16(2):111–139.

Touretzky, D. S. 1986. *The Mathematics of Inheritance Systems*. San Francisco, CA, USA: Morgan Kaufmann.

Varshney, L. R.; Pinel, F.; Varshney, K. R.; Bhattacharjya, D.; Schörgendorfer, A.; and Chee, Y.-M. 2013a. A big data approach to computational creativity. arXiv:1311.1213v1 [cs.CY].

Varshney, L. R.; Pinel, F.; Varshney, K. R.; Schörgendorfer, A.; and Chee, Y.-M. 2013b. Cognition as a part of computational creativity. In *Proc. 12th IEEE Int. Conf. Cogn. Inform. Cogn. Comput. (ICCI*CC 2013)*, 36–43.

Wasko, M., and Dale, R. 1999. Generating natural language descriptions of project plans. In *Advanced Topics in Artificial Intelligence*, 109–121. Berlin: Springer.

Wu, J.; Yildirim, I.; Lim, J. J.; Freeman, B.; and Tenenbaum, J. 2015. Galileo: Perceiving physical object properties by integrating a physics engine with deep learning. In *Adv. Neural Info. Process. Syst. 28 (NIPS 2015)*. 127–135.

Zhong, Z., and Ng, H. T. 2010. It makes sense: A wide-coverage word sense disambiguation system for free text. In *Proc. Assoc. Comput. Linguist. Annu. Meet. Syst. Demonstr. (ACLDemos'10)*, 78–83.

Live, Die, Evaluate, Repeat:
Do-Over Simulation in the Generation of Coherent Episodic Stories

Stefan Riegl
Institute of Cognitive Science
University of Osnabrück
D-49069 Osnabrüuck, Germany
sriegl@uni-osnabrueck.de

Tony Veale
School of Computer Science and Informatics
University College Dublin
Belfield D4, Ireland
tony.veale@ucd.ie

Abstract

Struggling writers are sometimes tempted to throw away their current effort and start over with a a blank page and a fresh approach. But the cost of yielding to this temptation is high, especially when one has already sunk a great deal of time and energy into a work that must be discarded. However, as computational creativity increases the speed and lowers the cost of narrative generation, the option of a fresh do-over becomes ever more attractive. So we consider here a simulation-based approach to the generation of episodic stories in which stories are generated, evaluated and frequently discarded in a rapid, coarse-grained cycle of engagement and reflection. The goal of simulation is to better exploit the situated possibilities for information transfer amongst the characters in a story, while the goal of *repeated* simulation is to find the story that achieves maximal coherence amongst its episodic parts.

Introduction

A compelling story is like the juiciest gossip, so it is likely that people have been sharing views about what constitutes a good story long before Aristotle ever wrote the *Poetics*. As with gossip, *how* we are told is as crucial as *what* we are told, and linguistic framing is as important as the events that make up the causal substance of the story. But in addition to stylistic subtleties, listeners also appreciate how the presentation of a story adapts to the constraints imposed by its medium. How did an ancient Greek bard arrange the narration of the *Iliad* to efficiently hold the audience's attention span? How did George Martin portion the sweeping *A Song of Ice and Fire* into books of several hundreds of pages each, or screenwriters parcel it into episodic films of an hour apiece, while maintaining continuity and coherence throughout?

Long stories are often divided into episodic chunks to facilitate distribution and consumption, but each episode must be relatively self-contained while coherently linking to what has gone before and what will come next. Episodes can benefit from a unifying theme, such as a common goal, antagonist or location, yet each must slot into the grander sweep of the narrative by echoing past events or foreshadowing future ones. The use of echoing and foreshadowing, as reflected in what characters say and do, creates coherence in what might otherwise seem a rambling sequence of disjoint events. So a character that commits an egregious act in one episode may be punished or blackmailed for it in the next, while others may alter their views on the basis of this knowledge. In this way, information-sharing across episodes lends purpose to action and unites episodes into a coherent whole.

When a story is coherent and its actions well-motivated, certain elements can be usefully left unsaid, as these gaps will be filled by an engaged listener (Abbott 2008). But which actions are the key-frames of a story and which can be interpolated between them? We argue that it is the events that promote future information transfer (gossip, blackmail, boasting, threatening, etc.) that are key to understanding the mindset of a story's characters (Owens, Bower, and Black 1979). The stories that specify actions to which other characters visibly relate are the narratives that listeners can relate to also. The goal of episodic story generation should thus be to maximize the opportunities for a narrative to create coherent relations between characters and with the audience.

We present here a simulation-based approach to story generation that is based principally upon the *Scéalextric* model of (Veale 2017), but which also integrates elements of the *engagement and reflection* cycle as identified by (Sharples 1999) and implemented by (Pérez and Sharples 2001) in the MEXICA system. *Scéalextric Simulator* [1] generates a series of short self-contained episodes that it links together to form a long over-arching story. Episodes can be chained in different ways, though popular convention suggests that a persistent main character is the best way to create a single narrative thread. Even when one character – a hero – persists, episodes must cross-relate in other ways too, so that actions are seen to have far-reaching consequences. *Scéalextric Simulator* uses repeated generation, simulation and evaluation to find the threaded sequence of local episodes that maximizes a global measure of information coherence.

Many different genres of stories exist in the wild, of which some impose less stylisitic constraints than others, like some poems that derive their charm from their lack of a specific form. For the stories presented here we follow the tradition of a structuralist view of stories, implying a form of narratological events that supports coherence in a story.

Related Work

Creative writing is a pasttime that has been practiced for millenia by novices and professionals alike. The sheer diversity

[1] http://afflatus.ucd.ie/simlextric/

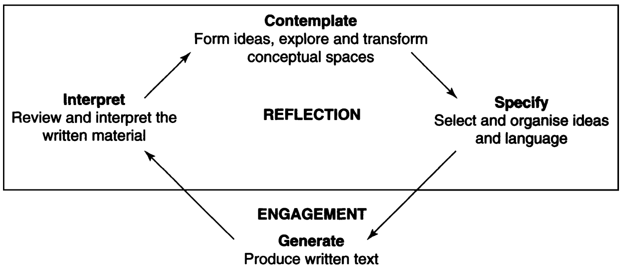


Figure 1: The writer's cycle of engagement and reflection, from (Sharples 1999).

of expression makes the process of creative writing particularly difficult to understand. The accumulated wisdom about the writer's craft, such as popular guides and maxims (e.g., "a story has a beginning, middle and an end"), often serve as metaphors or as simplified ideals, and rarely contribute to a deeper understanding of the act of writing itself.

Sharples (1990) has championed the view that writing is a process of problem-solving and creative design. He argues that writers do not start with a single intention but with a set of constraints that frames a space of possible solutions. Such constraints are limiting, but they usefully tell an author where focus can most effectively be placed. Even when making a humble shopping list, writing is more than the physical act of etching words on paper; it requires cognitive effort to evaluate an emerging text – what has been written and what remains to be said – and to satisfy the constraints of the given task. Sharples argues that these two processes occur sequentially in an alternating fashion (see Fig. 1). An engaged author continues to produce text until a cause, external or internal, necessitates a moment to stop and reflect. During reflection, the existing text is reviewed, new ideas are conceived, and a selected few are prepared for integration into the text, at which point engagement can continue.

The MEXICA model of (Pérez and Sharples 2001) gives an implemented, computational form to Sharples' model of engagement-reflection. MEXICA was designed to automate the generation of short stories and thereby facilitate the study of human creativity. It builds each new story from a stock of predefined story-structures, a set of adjustable parameters, and a list of explicit external and internal constraints. MEXICA operates with an inventory of primitive story actions, and a memory of previous stories composed only of those actions; those past stories illustrate how actions combine to form coherent plot sequences. In line with (Sharples 1999), MEXICA alternates between an engagement and a reflection phase: during engagement, new and appropriate actions are appended to the emerging story in a way that obeys rhetorical and thematic constraints. If the logical consequence of a new action contradicts the emerging story context, the story become incoherent and the system begins to reflect on why this is so. The emerging story is assessed in terms of novelty, interestingness and coherence, and appropriate actions are chosen for insertion into the story to restore coherence.

Coherence is a central concept in story generation, and one that operates over – and thus varies with – the stock of elements from which a story is to be constructed. For instance, early in the 20th century the Russian structuralist Vladimir Propp created a system to formalise Russian folk tales as sequences of character functions (Propp 2010). A text may contain one or more tales, of which each contains one or more *moves* that each permit a more-or-less self-contained narration. Each move comprises a sequence of functions, of which Propp defines 31 for Russian folk tales. Sequential moves can be interleaved in a variety of ways and exhibit dependencies of varying strength to each other. So when a later move introduces a new villain, the hero established in an earlier move will naturally reappear. Coherence in the Proppian scheme requires an author to ensure that moves are connected in ways that satisfy their associated character functions (Gervás 2013). The sequence of moves in a tale can be reordered provided the connections between functions remains satisfied.

A landmark in the history of computational storytelling is TALE-SPIN (Meehan 1977), a system that generates diverse stories from initial character descriptions or a desired moral. The movements of TALE-SPIN's characters through its story world are simulated so that they perform their actions rationally at each juncture, to realize their predefined goals. The sequential movements of a focal character are then arranged to provide the rendered story. TALE-SPIN's simulator uses knowledge of multiple domains to infer character beliefs, to resolve goals, and to translate goals into actions. TALE-SPIN is a spinner of short tales and no attempt was made to divide a narrative into a series of episodes or Proppian moves. In each state of the simulated world, the next is derived by allowing characters to perform their selected plans of action, and it is the rationality of these goal-based actions that imbues the story with a coherent shape.

The Knowledge-Intensive Interactive Digital Storytelling (KIIDS) model of (Gervás et al. 2005) is a framework for story generation that offers a set of ontologies modeling knowledge (and *meta*-knowledge) of the interactions in stories. Built on this is ProtoPropp, an automated story generator that gives Propp's analysis a computational form. ProtoPropp's stories typically feature multiple episodes that are created sequentially, where case-based reasoning selects the next episode based on constraints explicated by "the current state of the narration and using explicit knowledge about narrative and world simulation" (Peinado and Gervás 2006). When ProtoPropp's performance was evaluated in a user study, judges were asked to rate the coherence and novelty of ProtoPropp's stories amongst a set of randomly generated stories and human stories taken from a corpus. Overall, ProtoPropp's stories were deemed to be considerably closer to the human stories then to the random-generatd ones.

Generating and Integrating Episodes

A coherent episodic narrative is built from locally-coherent chapters (or *episodes*) that unite to maximize a global objective function. The *Scéalextric Simulator* presented here uses the *Scéalextric* system to generate the individual episodes which it then stitches together, simulates, and evaluates. As in *Scéalextric*, we assume that each episode is a tale of two characters (Veale 2017), but at least one of these can vary from episode to episode to yield a narrative with potentially many characters. *Scéalextric* is used to generate episodic plots with generic characters A and B, and *Scéalextric Simulator* then decides how these placeholders are to be filled.

Generating Episodes

(Veale 2017) modeled a two-character plot as a random-walk in a forest of causal links between action verbs. Each verb relates an A to a B, and sucessive verbs typically shift the focus from A to B and back to A. Bookend texts are defined for each verb so that a story can meaningfully begin with any action and terminate at any action. If we view the causal forest as a search-space, a coherent path can often be found between any two actions. Using the *Flux Capacitor* of (Veale 2014) to suggest actions at the beginning and at the end of a character's journey through a stereotypical category – e.g. going to medical school and losing one's medical license sit at opposite ends of one's journey through the Doctor category – *Scéalextric Simulator* can select meaningful start and end actions for the focal character in an episode, and use *Scéalextric* to fill in the rest of the plot. *Scéalextric* also provides ample templates for the rendering of plot actions in idiomatic forms, as well as causally-pairwise connections (such as *but, then, so*) to link plot actions.

There are as many episodes as there are character arcs, since each episode represents the journey of a character through a certain category, say from *pauper* to *millionaire* or from *believer* to *apostate*. To speed up story generation itself, we pre-generate a large inventory of these episodes – 12,000 in all – to be assembled into long-form narratives by the system. This inventory comprises alternate pathways through the causal forest for each of the character arcs defined in (Veale 2014). It allows *Scéalextric Simulator* to focus on the simulation of the plot in each episode rather than on its generation, and on the integration of multiple episodes into a coherent whole.

Linking Episodes

During simulation, the system maintains a placeholder view of characters. They are defined by their actions in the story, not by any prior knowledge. As characters interact in an episode, they gain information about each other. Interaction may involve issuing a threat, making a promise, or sharing a story about a past event. Over time, characters learn the things that make other characters proud or shameful, and can exploit this information to advance their own goals. Unlike TALE-SPIN and subsequent planning systems (e.g. (Riedl and Young 2010), these goals are not advanced incrementally; recall that the plot structure for each episode is generated prior to simulation. Rather, when an action is performed by a character as per the plot, and that action allows acquired information to be exploited, it is rendered as such, and the system records its contribution to global coherence.

Scéalextric Simulator provides such a view. For each of the 800+ A-B actions defined by *Scéalextric*, we associate at least one *continuation* action that an observer C may be likely to perform on A as a result. Thus, when episode n concludes with the event $A\ V_1\_act\_on\ B$ then a follow-up episode $n+1$ is chosen on the basis that its first action, $C\ V_1\_act\_on\ A$ is a valid continuation, inasmuch as $V_1\_act\_on$ suggests $V_2\_act\_on$ in the three-body causal model. There are episodes for any starting action, and each action is associated with multiple continuations. The selection of follow-on episodes is thus non-deterministic, and the simulator selects a follow-on randomly from the available choices.

Creating a World of Feelings and Memories

The *Scéalextric Simulator*'s knowledge-base describes characters, their actions, their locations and their beliefs. Each character resides at one location at a time, and can move between locations at episode boundaries. Each possesses a set of *affective* beliefs, each of which concerns a past action known to the character, as well as the intensity and type of the character's "feeling" toward it. Intensity, an integer, ranges from 1 to 9, while type can be *proud*, *guilt*, *admiration* or *shock*. Agents feel pride or guilt for their own actions and admiration or shock for the actions of others. If an agent feels *pride* for its own action, an observer will feel *admiration* for that action, while if the agent feels *guilt* for its own actions, others will feel a degree of *shock*. Information is gained by characters either by observing their own actions or the actions of others. All 800+ of Scéalextric's actions have been assigned at least one pairing of type and intensity.

(Veale and Valitutti 2017) describe how a rich inventory of famous characters, called the NOC list, can be integrated into the story-generation process, so that their attributes and proclivities are reflected in the story's rendering. We do not exploit this depth of prior character detail here, and use the NOC list only to suggest the names of story characters (e.g., Bill and Hillary, Tom and Jerry, etc.). Character names are assigned to placeholders in the narrative (A, B, C, etc.) after the episodic plot structure has been created. At this point, each episode is associated with a different locale, drawn from a range of vivid options in the NOC list (e.g., a seedy nightclub, a ritzy hotel lobby).

The protagonist (denoted A) is the character that persists across all episodes. All other characters are antagonists (denoted B, C, etc.). For all antagonists a "common past" is generated as a collection of shared beliefs. For a number of iterations of the simulator, a variable number of antagonists (but at least two) are assigned to a temporary, virtual location. Two characters from this virtual location are chosen to participate in a randomly-generated plot structure. Each action from the plot structure is simulated, which allows all the characters in the virtual location to observe and affectively react to the action, adding new beliefs to their memories.

Building on this foundation of shared memories, episodes are incrementally added to the emerging story. The first is chosen randomly, and subsequent episodes connect to the last via a causal continuation of its final action. Each new episode is set in a new locale and introduces a new antagonist. Glue text is inserted between episodes to explain the change of locale, and before the first episode and after the last episode to frame the story as a whole. In rare cases, a story cannot be progressed because no continuation can be found to launch a new episode. In these cases the simulation ends early, and the failed story is punitively scored.

Simulate, Reflect, Repeat.

In the MEXICA system of (Pérez and Sharples 2001) the unit of engagement is the action. New actions are added to an evolving tale in a process of engagement, prompting reflection on the consequences of each addition. The unit of engagement in the *Scéalextric Simulator* is not the action but the episode. New episodes are added to an evolving tale in a process of engagement, prompting reflection on their contribution to information transfer and global coherence.

A key issue concerns *when* reflection should take place. Should it occur incrementally, after the addition of each new episode, or upon the completion of a story? MEXICA employs incremental engagement, insofar as it pursues a greedy approach to generation: an evolving story is worked and reworked until it satisfies the desired constraints. Yet in a non-greedy approach that explores many alternate stories, it makes sense to reflect *after* each is completed and simulated. Scoring each story using an objective function that rewards global coherence, the highest-scoring story can be selected from a run of perhaps thousands of successive simulations.

Engagement in the *Scéalextric Simulator* governs the integration and simulation of new episodes into the story, while reflection governs the evaluation of each story once it has been completed. Engagement thus includes the transfer of information amongst characters, either by observing an action or reporting it to others. Reflection evaluates the impact of this transfer on the global coherence of the story.

Certain Scéalextric actions are defined as vectors of information transfer. Among others, these include *deceive*, *teach*, *confide_in* and *share_stories_with*. Deceitful communication requires the simulator to invent a false belief to communicate, while truthful communication (which is otherwise assumed) causes the most intense belief of the appropriate type to be transferred from sender to recipient (and all other observers). The action *confess_to* requires the type of the transferred belief to be one for which the speaker feels *guilt*. So when one character blackmails another, the extortion is assumed to relate to the most guilty feeling held by the victim. The coherence of a blackmail action is a function of the intensity of the guilty secret that one is blackmailed about, so a story in which A is blackmailed for killing B is preferred over one where A is blackmailed for merely insulting B.

This scoring of a story, to reward stories with well-motivated actions and punish those with weaker rationales, is a matter for the reflection phase. Whenever an action in one episode is motivated not just by the local plot, but by the actions of an earlier episode that are accessible due to information transfer, the overall story is rewarded accordingly. These connections across episodes also influence the rendering of the finished story, either through the insertion of mini-flashbacks, or by the rendering of direct speech that explicitly harks back to an earlier motivating action.

Once connections between episodes are established, and repeated simulation has identified a sequence of connected episodes to serve as a global plot, the plot is rendered as text. This rendering proceeds largely as in (Veale 2017) with some additions to create a more pleasant reading experience. Character names are rendered in different ways (first long, then short) or are replaced by gender-appropriate pronouns to avoid over-use of names. Adjectives that are suited to the respective action (e.g. *violent* for *attack*) are inserted into the idiomatic text to embellish the rendering of the protagonist and/or antagonist. Connecting text between episodes is inserted to note the movement of the persistent character between locales, and to establish the locale of the new episode.

An Objective Function for Global Coherence

The *Scéalextric Simulator* runs not one, but many simulations, one for each of its many successive attempts at generation. It scores each run according to an objective function that rewards global coherence, and renders the highest-scoring narrative into a polished idiomatic text. As an initial computational evaluation we quantify several features that influence the cross-episodic coherence of a story. The objective function $o(S)$ for the simulation S of a newly-generated narrative is defined as a summative score for the features in S divided by the square root of the number of actions in S:

$$o(S) = \frac{\sum_{f \in F(S)} s(f)}{\sqrt{\sum_{e \in S} |e|}}$$

Here $e \in S$ denotes an episode in the narrative under simulation in S and $|e|$ denotes the number of actions in the plot structure assigned to e by *Scéalextric*. $F(S)$ denotes the set of scorable features in S and $s(f)$ denotes the score for a single feature f, as defined in table 1. We divide by the square root of the total number of actions in a narrative to punish unnecessarily long stories whose actions do not earn their keep by contributing to the global coherence of the narrative. We divide by the square root because our empirical investigations show that the number of features does not tend to increase linearly with the length of a narrative.

Net Score	Feature Description
+10	information transfer (direct speech)
+6	– a character is lying
$+n$	– intensity n of the belief to be shared
+10	– the protagonist is talking
+20	– reference to past action
+50	– information shared for 2nd, 3rd, etc. time
+30	continuation of past action

Table 1: Relative contribution of features to story coherence.

The values in table 1 have been determined empirically over many trial simulations during feature development. The objective function thus rewards stories whose episodes are tightly cross-stitched by information transfers between characters. As noted earlier, these transfers are weighted by intensity and scored accordingly, since coherence is heightened when a narrative hinges on actions with pervasive influences that extend across episodes. So the actions that prove pivotal to a story, insofar as they motivate multiple future actions, are scored most generously of all. Actions that do not contribute to the score of a narrative are dead-weight, and repeated generation and simulation is a means of minimizing this narrative flab. This intuition is captured in the maxim

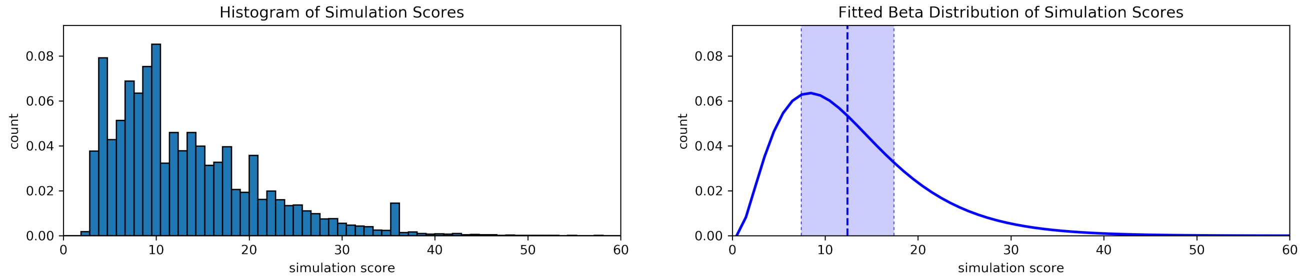


Figure 2: Distribution of global coherence score for 100,000 stories. A probability density function of a beta-distribution was fitted to the score distribution.

of "Chekov's pistol," which dictates that eye-catching flourishes (such as a gun prominently mounted on the wall) must earn their keep by meaningfully influencing the narrative.

Figure 2 presents the distribution of coherence scores produced by our objective function when we simulate 100,000 different three-episode narratives. The mean and standard deviation are 12.5 and 4.97 respectively, but we can greatly improve on the mean score with successive cycles of generation and simulation, retaining the highest-scoring narrative and discarding all others. To find a higher-scoring narrative, we simply run more cycles of generation and simulation. Figure 3 graphs the rate of increase in the score of the best narrative across repeated runs of generation and simulation. Each point in this graph represents the mean of the score for 100 three-episode stories. As also shown in Figure 3, a logarithmic function has been fitted to these mean values, demonstrating a logarithmic increase in mean score for repeated runs of the system. In addition to the mean for each iteration, Figure 4 also shows the standard deviation for each successive iteration across all stories.

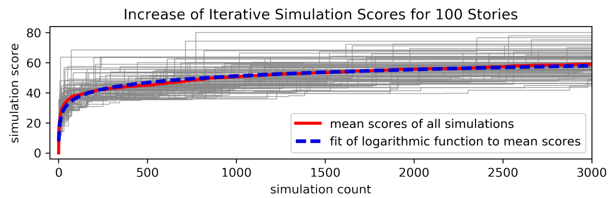


Figure 3: The changing scores of simulations for 100 generated stories. Mean values are per iteration over all stories and approximate a logarithmic function.

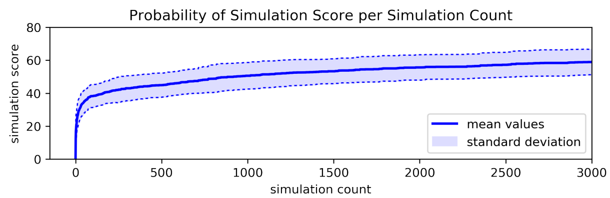


Figure 4: The changing scores of simulations for 100 generated stories, with mean values and standard deviation per iteration over all stories.

But how much is ever enough? When should the generator by satisfied with a particular narrative and a particular score? To achieve excellence we must quantify excellence, by e.g. imposing explicit minimum scores that must be exceeded by a qualifying narrative. For instance, as shown in Figure 3, 1000 cycles of generation, simulation and evaluation – which requires just seconds to execute – is typically sufficient to achieve a score that is multiple standard-deviations higher than the mean score achieved for any given narrative. Alternately, we can express this threshold as the number of standard deviations above the mean that is required for excellence. A *six sigma* threshold thus requires a successful narrative to score more than six standard deviations above the mean on the system's measure of global coherence. Or we can do without thresholds altogether, and simply use the available time to the fullest. In a *just-in-time* setting, the system generates new narratives for as long as it is permitted. When the system is interrupted, or its allotted time has run out, it simply returns the highest-scoring narrative it has thusfar generated.

Evaluation

We have seen that repeated do-overs tend to steadily increase the scores achieved by narrative generation on our objective function, which has been defined to codify our own notions of global narrative coherence. But does our notion of coherence comport with the intuitive view held by others, such as by the end consumers of the stories that are generated? What prospect is there of asking lay judges to rate the "coherence" of a long narrative in a way that facilitates meaningful empirical evaluation?

(Veale and Valitutti 2017) present the results of crowd-sourcing experiments in which anonymous judges were asked to rate stories generated by *Scéalextric* under two conditions and along six dimensions: *eventfulness, imagination, laughter, silliness, entertainment* and *vividness*. Stories from *Scéalextric Simulator* were evaluated according to these six dimensions using the same approach and the same testing conditions as (Veale and Valitutti 2017), again using the CrowdFlower platform with comparable demographics for the 50 test subjects a.k.a. judges. While the mean results (of 10 judgments per story per dimension) showed meaningful

The story of Maura

Maura needed a place to live and Saoirse had plenty of it.

Maura found Saoirse at an underground lair. Maura rented accommodation from her. She paid Saoirse what she owed. Saoirse could not achieve bossy Maura's lofty goals. She refused to honour Saoirse's commitments to her, so Maura ripped off rich Saoirse's best ideas. Anguished Saoirse evicted Maura from Saoirse's home.

At a smoke-filled back room Maura met Oscar Wilde. Maura assiduously curried favor with dictatorial Oscar after cheated Saoirse evicted Maura from Saoirse's home. Maura told eager Wilde a pack of lies. Maura said: "Dolores wrote propaganda to promote your cause." His attitude hardened toward Maura. He openly disrespected Maura because earlier she took everything that Saoirse had. Maura tried to tune out loudmouthed Oscar's voice. Bossy Oscar Wilde wrote Maura off as a loser, so he coldly dismissed Maura and turned away.

It was at the red carpet when Maura found Rina. Maura started a new job for influential Rina after unsatisfied Oscar told Maura to get out and not come back. Rina took full advantage of her. She pulled the wool over Maura's eyes. She said: "Saoirse was a real suck-up to aristocratic Wilde." Maura could not reach the bar set by bossy Rina. She was very disappointed in her, so "Get out! You're fired" said Rina.

It was at a recording studio when Maura found Dolores. Authoritarian Dolores recruited Maura into her ranks after Rina asked her to clear out her desk and leave. Maura took the spotlight from lackadaisical Dolores. Dolores withheld due payment from lazy Maura. Maura criticized sinful Dolores in public. She said: "Saoirse showed no shame in sucking up to influential Wilde." She broke with her and went her own way.

What do you think? Can Maura and Dolores ever mend their relationship?

Figure 5: An full example story generated by the Scealéxtric Simulator. The story has four episodes with a variable number of sentences, an opening and closing sentence and episode intros. Direct speech is used to present transfer of information and inserted clauses at the end of sentences refer to motivating actions from previous episodes.

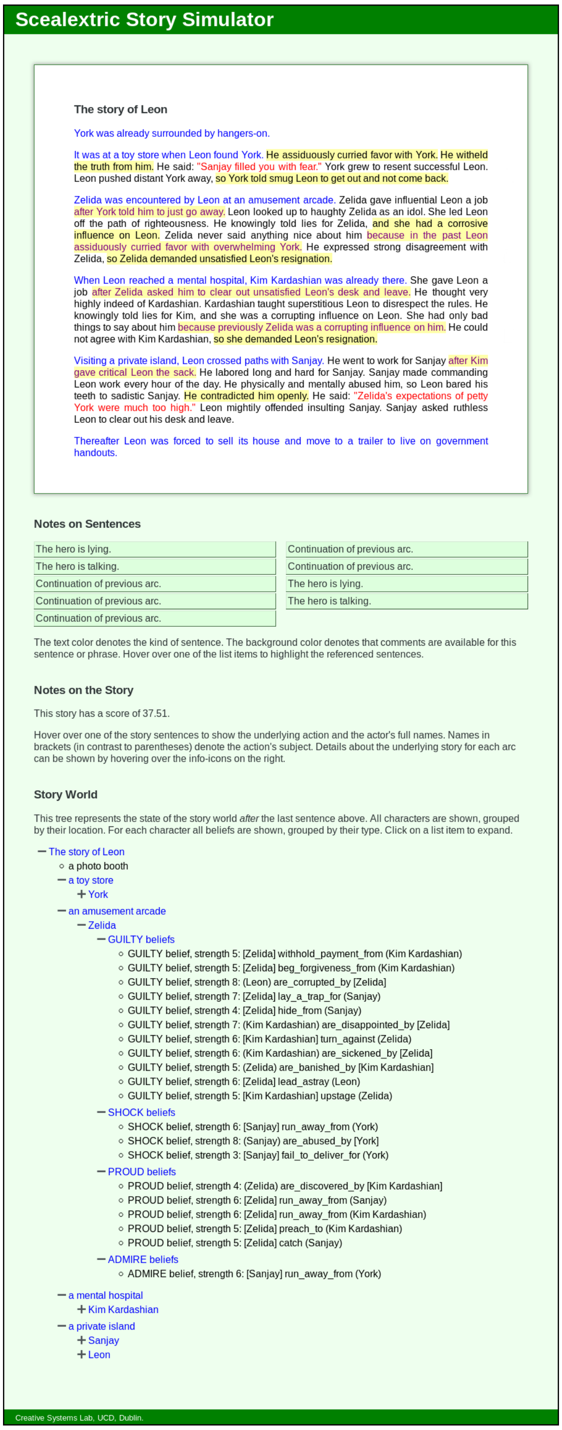


Figure 6: An excerpt of a story in the web interface as of February 28, 2018. Introspection tools can be used to analyse the presented story, including tool tips and colour-coded annotation of the story text, interactive highlighting of sentences and clauses that increased the score and a hierarchical representation of locations, characters and beliefs. The web interface is accessible here: `http://afflatus.ucd.ie/simlextric/`

separation between the two conditions, it was also ruefully noted that many judges opt for the middle rating when presented with a Likert scale. Given that the stories evaluated in (Veale and Valitutti 2017) were short single-episode affairs, we can expect more judges to take the lazy middle option when presented with multi-episode narratives that are considerably longer and more taxing to analyse.

This was indeed the case when we replicated those earlier experiments on three-episode narratives generated by the *Scéalextric Simulator*. Although plots have now been given a greater sense of direction through the use of *Flux Capacitor* to specify the start and end actions of a causal path and thereby create a more meaningful arc for the protagonist, and episodic plots are integrated using 1000 cycles of simulation to select the highest-scoring narratives, the mean judgments of anonymous human raters tend to be lower than either condition in the original experiments. For the most part, the new results fall within a standard deviation of the old, with one notable exception: the *laughter* dimension.

The two conditions in (Veale and Valitutti 2017) relate to humorous intent. In the simple generic condition, straight narratives – which is to say, narratives that are not intended to be humorous – are generated using baseline *Scéalextric* capabilities. In this generic condition, A and B character placeholders are simply filled with random animals in the Aesop tradition (e.g. the monkey and the snake). In the NOC condition, A and B are chosen to be familiar characters that are, in a deliberate metaphorical sense, well-matched. These characters are further chosen so as to evoke a meaningful incongruity in the guise of postmodern irony. Thus, fictional characters are paired with real people, as in Lex Luthor and Donald Trump, or characters are paired with similar entities from different eras, as in Steve Jobs and Leonardo Da Vinci, or characters are paired on the basis of a shared screen portrayal, as in Frank Underwood and Keysar Söze. In this condition, actions are rendered into text using vivid details of the characters as provided by the NOC. The goal is to foster humorous incongruity in character and action while using the same basic plotting mechanism of the generic condition.

Narrative coherence is no laughing matter. All things being equal, we can expect a long narrative with low internal coherence to strike a reader as more laughable – perhaps more laughably bad – than one whith high internal coherence. Most contemporary theories of humour thus emphasize the role of incongruity in the construction of laughter-inducing texts (see e.g., (Suls 1972), (Raskin 1985), (Ritchie 1999), (Ritchie 2003)). An incongruity is any logical impasse or jarring misalignment of expectation and reality that stops readers in their tracks. Some incongruities are more subtle than others, while others are significantly more dramatic. A vexing question surrounds the actual role of incongruity in humour: is it a profound phenomenon that tickles the funny bone and triggers cognitive recovery mechanisms, or is it merely an epiphenomenon that accompanies, but does not explain, most instances of humour (Veale 2004)?

In any case, laugher is a visceral response to a situation. Unlike the other dimensions evaluated in (Veale and Valitutti 2017), which each need a brief explanation so that judges can understand what they are supposed to score, laughter needs no explanation. Narrative coherence, quite unlike laughter, is a rather abstract quality that is far from visceral, and requires an even more substantive explanation to judges than any dimension in the original experiments. However, insofar as low coherence creates the conditions for unintended incongruities to arise in a text, we can expect a straight narrative with low coherence to evoke more laughter on average than a straight narrative with high coherence.

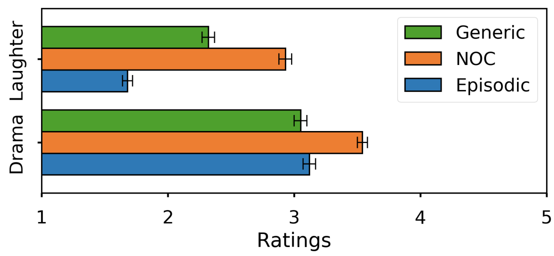


Figure 7: Comparison of mean ratings for the dimensions of drama and laughter for the NOC and Generic conditions from (Veale and Valitutti 2017) and the Episodic condition from this evaluation. Error bars denote standard errors.

The experiments reported in (Veale and Valitutti 2017) show that humour can be engineered in a text by fostering the kinds of incongruity – between real and fictional, or contemporary and historical, or between distinct fictional worlds – that encourage laughter. Those earlier experiments report a mean Laughter score for NOC stories of 2.93, significantly higher than the mean Laugher score for generic stories, 2.32. In effect, the narratives that were engineered to be creatively incongruous were deemed to be significantly more humorous than their straight counterparts.

Given that our multi-episode narratives are *not* engineered to be humorous, and are, moreover, engineered to be as internally consistent and free of incongruity as possible, we expect the mean laughter scores for these narratives to not only fall far short of those attained for NOC narratives, but to also fall significantly short of those attained for generic narratives. As shown in Figure 7 this is indeed the case. With a mean score of 1.68 for laughter, our episodic narratives have seemingly been drained of their inconsistencies by repeated engagement and reflection, so that they prompt much less unintended laughter than comparable stories that are not chosen for their coherence.

From Meet-Cute to Cliff-Edge and Beyond

The flow of information between the characters in a story – who knows *what*, and *how/why/when* do they know it? – is every bit as as important as the flow of information from author to audience. For how this flow is managed will dictate the coherence of the narrative and influence the feelings it engenders in a reader. For example, as shown in (Delatorre et al. 2017), a tightly-managed information flow can greatly enhance the enjoyable sense of suspense that authors hope to nurture in consumers of thrilling or mysterious stories. In this work we have focused more on the feelings of story characters than on story consumers, in the hope that the latter will feel the benefits of coherence that accrue from the careful tracking of the former. This is especially important for the generation of longer stories that incorporate multiple episodes, characters and locales.

We have shown that repeated simulation of what characters know, how and when they know it, and how they exploit this knowledge to advance the plot, underpins a measure of global coherence that can be steadily increased over repeated cycles of generation and iterative evaluation of episodes. While this approach departs substantially from how humans create stories, we believe it can nonetheless be considered a coarse-grained version of the engagement-reflection loop that is championed by (Sharples 1999) and implemented in the MEXICA system of (Pérez and Sharples 2001). It is an approach that makes a virtue of starting over, of failing fast and of failing better, because in conditions like ours it is more costly to fix a broken, highly-constrained episode than to make a fresh start from the last known good episode. Moreover, it facilitates a *just-in-time* view of the story-generation process that is ideally suited to the implementation of that process as a creative web service (see e.g., (Veale 2013) and (Concepción, Gervás, and Méndez 2017)).

The crowd-sourced evaluation has been conducted as a pilot for a more comprehensive study yet to come. Vexing challenges with the evaluation of long stories have forced us to look for validation of our objective function via a roundabout and creative interpretation of the experimental results. Understandably, anonymous raters who are paid small amounts per rating cannot be trusted to fully engage or to give a reliable picture of anything but the most visceral of phenomena. As we improve our objective function to capture additional aspects of global coherence, we will have to find other means of evaluating the resulting stories.

As shown in (Delatorre et al. 2017), a viable area of improvement concerns the fostering of suspense at the boundaries of adjacent episodes. In the "cliff-hanger" serials of old, in which long cinematic narratives were broken into a series of weekly instalments, each episode would conclude with a moment of high suspense by placing the protagonist in a position of impending doom. Subsequent episodes would quickly deflate the suspense, only to create a new predicament for the hero to endure. Long stories that manage the ebb and flow of suspense in this way should do more than hold the interest of an engaged reader: they may also go some way toward holding the sustained attention of an otherwise disinterested crowd-sourcing volunteer.

References

Abbott, H. P. 2008. *The Cambridge introduction to narrative*. Cambridge University Press.

Concepción, E.; Gervás, P.; and Méndez, G. 2017. An api-based approach to co-creation in automatic storytelling. In *6th International Workshop on Computational Creativity, Concept Invention, and General Intelligence. C3GI*.

Delatorre, P.; León, C.; Salguero, A.; Mateo-Gil, C.; and Gervás, P. 2017. Impact of interactivity on information management for suspense in storytelling. In *Proceedings of the 4th AISB Symposium on Computational Creativity*. AISB.

Gervás, P.; Díaz-Agudo, B.; Peinado, F.; and Hervás, R. 2005. Story plot generation based on cbr. *Knowledge-Based Systems* 18(4-5):235–242.

Gervás, P. 2013. Propp's morphology of the folk tale as a grammar for generation. In *OASIcs-OpenAccess Series in Informatics*, volume 32. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik.

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

Owens, J.; Bower, G. H.; and Black, J. B. 1979. The "soap opera" effect in story recall. *Memory & Cognition* 7(3):185–191.

Peinado, F., and Gervás, P. 2006. Evaluation of automatic generation of basic stories. *New Generation Computing* 24(3):289–302.

Pérez, R. P. Ý., and Sharples, M. 2001. Mexica: A computer model of a cognitive account of creative writing. *Journal of Experimental & Theoretical Artificial Intelligence* 13(2):119–139.

Propp, V. 2010. *Morphology of the Folktale*, volume 9. University of Texas Press.

Raskin, V. 1985. *Semantic Mechanisms of Humor*. D. Reidel.

Riedl, M. O., and Young, R. M. 2010. Narrative planning: balancing plot and character. *Journal of Artificial Intelligence Research* 39(1):217–268.

Ritchie, G. 1999. Developing the incongruity-resolution theory. In *Proceedings of the AISB Symposium on Creative Language: Stories and Humour*. AISB.

Ritchie, G. 2003. *The Linguistic Analysis of Jokes*. Routledge Studies in Linguistics, 2. Routledge.

Sharples, M. 1999. *How we write: Writing as creative design*. Psychology Press.

Suls, J. 1972. A two-stage model for the appreciation of jokes and cartoons: An information-processing analysis. In Goldstein, J. H., and McGhee, P. E., eds., *The Psychology of Humor*. New York, NY: Academic Press. 81–100.

Veale, T., and Valitutti, A. 2017. Tweet dreams are made of this: Appropriate incongruity in the dreamwork of language. *Lingua* 197:141–153.

Veale, T. 2004. Incongruity in humor: Root-cause or epiphenomenon? *HUMOR: The International Journal of Humor* 17(4):419–428.

Veale, T. 2013. A service-oriented architecture for computational creativity. *Journal of Computing Science and Engineering* 7(3):159–167.

Veale, T. 2014. Coming good and breaking bad: Generating transformative character arcs for use in compelling stories. In *Proceedings of ICCC-2014, the 5th International Conference on Computational Creativity, Ljubljana, June 2014*.

Veale, T. 2017. Déjà vu all over again: On the creative value of familiar elements in the telling of original tales. *In Proceedings of ICCC 2017, the 8th International Conference on Computational Creativity, Atlanta, Georgia, June 19-23*.

"Talent, Skill and Support." A Method for Automatic Creation of Slogans

Khalid Alnajjar
khalid.alnajjar@helsinki.fi

Hadaytullah Hadaytullah
hadaytullah.hadaytullah@helsinki.fi

Hannu Toivonen
hannu.toivonen@helsinki.fi

Department of Computer Science and HIIT, University of Helsinki, Finland

Abstract

Slogans are an effective way to convey a marketing message. In this paper, we present a method for automatically creating slogans, aimed to facilitate a human slogan designer in her creative process. By taking a target concept (e.g. a computer) and an adjectival property (e.g. creative) as input, the proposed method produces a list of diverse expressions optimizing multiple objectives such as semantic relatedness, language correctness, and usage of rhetorical devices. A key component in the process is a novel method for generating nominal metaphors based on a metaphor interpretation model. Using the generated metaphors, the method builds semantic spaces related to the objectives. It extracts skeletons from existing slogans, and finally fills them in, traversing the semantic spaces, using the genetic algorithm to reach interesting solutions (e.g. "Talent, Skill and Support."). We evaluate both the metaphor generation method and the overall slogan creation method by running two crowdsourced questionnaires.

Introduction

Rhetorical devices are ubiquitous, they are used in daily communications, news, poems, and advertising. This paper focuses on slogans; more specifically, it tackles the task of creating slogans computationally. Slogans are memorable short phrases that express an idea about a product, and are commonly used in advertising campaigns.

In advertising, it is essential to construct expressions wisely. A research conducted by Reece, Van den Bergh, and Li (1994) suggests that recalling a slogan relies mainly on the slogan itself, not on the advertising budget, years in use or themes. Constructing such novel and interesting expressions is a time-consuming task for humans and a challenging one for computers. The method proposed in this paper aims at facilitating the process of constructing such creative expressions by suggesting inspirational slogan candidates tailored to user's desire. As a result, creative professionals (e.g. writers, advertisers, etc.) can collaborate with computers to produce creative results more efficiently.

Rhetorical devices in slogans have different effects on consumers (Burgers et al. 2015). In this paper, inspired by the work of Miller and Toman (2014), we focus on the two most common rhetorical devices found in slogans: (1) metaphors and (2) prosody. Miller and Toman have analysed 239 slogans and discovered that 92% of them contained at least one rhetorical device. Tom and Eves (1999)'s research has found that slogans containing rhetorical devices are more persuasive and have higher recall than those that do not.

Our method accepts a target concept and an adjectival property as an input. In advertising, the target concept and adjectival property would be the product type (e.g. a car) and the desired property that the slogan should express (e.g. elegant or luxurious). The method commences by generating apt metaphors for attributing the input property to the target concept. Thereafter, it creates expressions, slogans in our case, adapted to the input and the generated metaphors. A genetic algorithm is employed in the method to search for interesting slogans in the space of possible solutions.

Metaphors consist of two concepts, a tenor and a vehicle following Richards (1936) terminologies, where some properties get highlighted or attributed to the tenor from the vehicle. For instance, in the nominal metaphor "Time is money", *valuable*, a property of the vehicle *money*, is highlighted in the tenor, *time*. In this paper, the process of metaphor generation targets producing suitable vehicle candidates for expressing the intended adjectival property while considering the input concept.

We also examine the effect of using a corpus-based metaphor interpretation model in generating metaphors. Moreover, we argue that slogans with balanced features (e.g. relatedness to the input and metaphoricity) are comparatively more creative than those with a single dominating feature.

The remainder of this paper is structured as follows. We first briefly review the related work on generating metaphors and rhetorical expressions. Thereafter, we give an overview of resources used by the method. We then describe the method for (1) generating metaphors and (2) generating slogans. Finally, we present the evaluations of our methods and discuss the results.

Related Work

In this section, we review the related work on two computational topics: (1) generation of metaphors and (2) generation of slogans and other creative expressions.

Generation of Metaphors

For the scope of this paper, we review two approaches for generating metaphors.

The first approach, by Xiao and Blat (2013), is focused on generating metaphors for pictorial advertisements. Their approach utilises multiple knowledge bases, e.g. word associations and common-sense knowledge[1], to find concepts with high imageability. The found concepts are then evaluated against four metrics, which are affect polarity, salience, secondary attributes and similarity with tenor. Concepts with high rank on these measures were considered apt vehicles to be used metaphorically.

Galvan et al. (2016) generated metaphors by using a web service, *Thesaurus Rex* (Veale and Li 2013), that provides categorizations of concepts and adjectival properties associated with them. Their approach starts by retrieving top 40% categories of the input tenor. It then selects an adjectival property, at random, that is associated with the tenor. Thereafter, it sends another query to the web service to obtain categories associated with the previously selected property. A category matching the retrieved categories of the tenor is selected. Finally, it creates a metaphor by finding a concept falling in the selected category which is also strongly associated with the selected property.

In contrast to the reviewed metaphor generation methods, our method employs a metaphor interpretation model to identify apt metaphors.

Generation of Creative Expressions

Strapparava, Valitutti, and Stock (2007) proposed a creative function for producing advertising messages automatically. Their approach is based on the "optimal innovation hypothesis" (Giora 2003). The hypothesis states that the optimal innovation is reached when novelty co-exists with familiarity, which encourages the recipient to compare what is known with what is new resulting in a pleasant surprise effect. The approach proposed by the authors utilizes semantic and emotional relatedness along with assonance measures to find interesting candidates of words to substitute some existing words in human-made familiar expressions.

Özbal, Pighin, and Strapparava (2013) have introduced a framework, *BrainSup*, for creative sentence generation. The framework generates sentences such as slogans by producing expressions with semantically related content to the target domain, emotion and colour, and some phonetic properties. The generated expressions must contain keywords that are input by the user. Using syntactical tree-banks of existing sentences as sentence skeletons and syntactical relations between words as constraints for possible candidate fillers, Özbal et al. have employed beam search to greedily fill in the skeletons with candidates meeting the desired criteria.

Using *BrainSup* as a base, Tomašic, Znidaršic, and Papa (2014) have proposed an approach for generating slogans using genetic algorithms instead of beam search. Moreover, their evaluation criteria were different from *BrainSup*'s evaluation. Tomašic et al.'s work demonstrated how it is possible to automatically generate slogans without any user defined target words by extracting keywords from the textual description of the target concept.

Regarding figurative language generation, *Figure8*, by Harmon (2015), generates metaphoric sentences. Five criteria were considered in the generation process, namely: clarity, novelty, aptness, unpredictability, and prosody. The system selects a tenor and searches for a suitable vehicle to express it. Thereafter, it composes sentences to express the metaphor by filling templates of metaphorical and simile expressions.

Our proposed method for generating expressions differs from existing methods as follows. It focuses on generating slogans for a product while expressing a single adjectival property. We want the property to be expressed indirectly and metaphorically. Furthermore, our method creates slogans whilst considering one skeleton at a time. Producing metaphorical expressions is addressed in *Figure8*, which in contrast is concentrated on similes.

Resources

This section covers the linguistic resources used in the proposed methods.

Corpus, ζ We use a 2 billion word web-based text corpus, *ukWaC*[2], as the main corpus. All corpus-based models in our approach are built using this corpus. We chose a web-based corpus to cover wide range of topics and different writing styles.

Language model, ξ We build a probabilistic bigram language model ξ using bigram frequencies provided with *ukWaC*. The language model is built to estimate the probability of a created slogan to be generated by ξ. A slogan with high probability is more likely to be grammatically correct as it appeared frequently in the corpus ζ. Employing bigrams, in contrast to trigrams or higher n-grams, gives the method a greater degree of freedom in its generations. Higher n-grams would improve the grammar of the generated expressions but would tie them to expressions in the original corpus.

Semantic model, ω We follow the approach described in *Meta4meaning* (Xiao et al. 2016) in building the semantic model ω. The goal of constructing this model is to find words that are semantically related to another word. We start by obtaining co-occurrence counts of words in ζ, constrained by sentence boundaries, within a window of ± 4. We limit the vocabulary of the model to the most frequent 50,000 words, excluding closed class words. We then convert co-occurrence counts to relatedness measure by employing the log-likelihood measure defined by (Evert 2008) while capping all negative values to zero. Finally, we normalize relatedness scores using L1-norm (McGregor et al. 2015).

[1] ConceptNet: `http://www.conceptnet.io`

[2] `http://wacky.sslmit.unibo.it`

Expression skeletons, δ A slogan skeleton is a parse tree of a phrase where all content words are replaced with a placeholder "***", i.e. stop words are kept. Nevertheless, all part-of-speech tags (e.g. VBZ) and grammatical relations (e.g. nsubj) between words are retained. The goal of using a database of skeletons is to reuse syntactical structures of effective slogans. The practice of reusing existing slogans can be observed in some well-known slogans, e.g. *Volkswagen*'s "Think Small." and *Apple*'s "Think Different.".

We utilise *Spacy*[3] as a natural language processing tool to parse 40 well-known slogans[4]. Prior to constructing the skeletons, we preprocess the obtained slogans to increase the parsing accuracy. The first preprocessing step is converting capitalized words into lower case, except the first word and any recognized named entities. This step reduces miss-classifying verbs as nouns, yet they could occur as many slogans are not complete sentences. Slogans tend to be informal; therefore, we convert words with the suffix *VERB*-in' into *VERB*-ing, in the second step. As a result of the preprocessing phase, *KFC*'s slogan "Finger Lickin' Good." becomes "Finger licking good.".

Subsequently, we convert slogans into skeletons. Figure 1 provides an example of a skeleton generated from *Visa*'s slogan "Life flows better with Visa.".

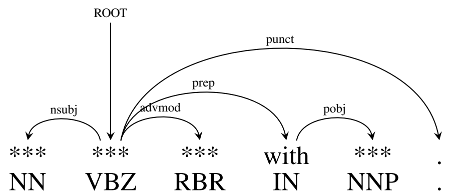


Figure 1: An example of a skeleton constructed from Visa's slogan: "Life flows better with Visa."

Once all slogans are transformed into skeletons, we only keep skeletons that have at least 40% of their tokens as placeholders and have a minimum of two placeholders. These conditions ensure that the method has some freedom in filling in the skeleton. As a result, slogans such as *Reebok*'s "I am what I am." and *Coca-Cola*'s "Enjoy." are removed. In total, the database contained 26 unique skeletons.

Grammatical relations, γ Similarly to approaches by Özbal, Pighin, and Strapparava (2013) and Tomašic, Znidaršic, and Papa (2014), we build a repository of grammatical relations. We parse the entire corpus ζ using *Spacy* and store all grammatical relations observed along with their frequencies. A grammatical relation contains a word (called dependent), its head word (called governor), the parts-of-speech of both words, and the type of relation. We retain grammatical relations with frequencies ≥ 50 to remove rare cases. The process yields 3,178,649 grammatical relations.

[3] http://www.spacy.io

[4] Obtained from: http://www.advergize.com

Nouns and Their Adjectival Properties, κ We employ two resources for retrieving nouns associated with the input property. The first resource, $\kappa_{General}$, is *Thesaurus Rex* (Veale and Li 2013). *Thesaurus Rex* is used for retrieving general nouns (e.g. coffee, flower, ...etc). On the other hand, the resource provided by Alnajjar et al. (2017), κ_{Human}, is employed to obtain nouns of human categories (e.g. actor, lawyer, politician, ...etc). These resources will be used in generating metaphors, the former for general metaphors and the later for personifications.

Method

In this section, we describe the proposed method. The input to the method is a target concept, T, and an adjectival property, P. An example of such input is $T =$"computer" and $P =$"creative".

The proposed method is broken into two processes, (1) metaphor generation and (2) slogan generation.

Generation of Metaphors

We define the metaphor generation task as follows. Given a tenor T and an adjectival property P, the generator produces vehicle candidates, $V = \{v_0, v_1, \ldots, v_i\}$. A vehicle highlights the adjectival property P in T when perceived metaphorically. An example vehicle candidate for expressing that a *computer* is *creative* is *poet*.

For the input property P, the method begins by retrieving nouns associated with P using κ. We retrieve two types of nouns from the resource κ, general nouns from $\kappa_{General}$ and nouns of human categories from κ_{Human}. We use the top 10% of each type to only pick candidates strongly related to P.

The above procedure gives nouns related to the given property P, but it does not ensure that their metaphorical interpretation in the context of tenor T is P. To select nouns that are likely to have the intended interpretation, we employ a corpus-based metaphor interpretation model, *Meta4meaning* (Xiao et al. 2016).

Meta4meaning accepts two nouns as input, a tenor and a vehicle, and produces a list of possible interpretations for the metaphor. To our knowledge, the proposed method here is the first for generating metaphors based on their interpretations.

Using *Meta4meaning*, the method interprets the potential metaphorical nouns retrieved by calculating the combined metaphor rank metric, c.f. Xiao et al. (2016). Only nouns with the property P among the top 50 interpretations are used. Additionally, as metaphors are asymmetrical, the approach removes vehicle candidates that have the interpretation rank of "T is [a] v" greater than to the interpretation of the reversed metaphor, i.e. "v is [a] T".

For example, nouns in κ that are strongly associated with $P =$ "$creative$" are:

$\kappa_{General}(creative)$ = {painting, music, ..., presentation}

$\kappa_{Human}(creative)$ = {artist, genius, poet, ..., dancer}

By interpreting these candidates using *Meta4meaning* and pruning out candidates not meeting the predefined conditions, we obtain the following candidates where the score is the interpretation rank:

$V_{General}(computer, creative)$ = {art: 4, drama: 4, director: 4, artist: 5, ..., exhibition: 50}

$V_{Human}(computer, creative)$ = {genius: 2, artist: 5, designer: 12, ..., inventor: 49}

Finally, we merge the two lists of potential vehicles into one, $V = V_{General} \cup V_{Human}$.

Generation of Slogans

The expression generation process takes the list of vehicle candidates V from metaphor generation process as input, as well as the initial input to the approach, i.e. T and P.

This section is divided as follows. We start by explaining how the semantic and search spaces which the method traverses are constructed. Thereafter, we motivate and define the aspects which we will consider while finding potential solutions, followed by a detailed description of generation algorithm.

Construction of Semantic Spaces From the pool of possible skeletons δ, the approach selects a skeleton s at random. Given a skeleton s, the method constructs two semantic spaces where words in them are used as potential fillers for s. These spaces are (1) interesting I and (2) universal Υ semantic spaces.

The interesting semantic space, which contains words that are favoured, is constructed by obtaining related words, from ω, to the input concept T and a vehicle v from list of vehicle candidates V. The method obtains the k words most strongly related to T. In our case k was empirically set to 150. The method includes related words to v to encourage the generation of metaphorical expressions. For any $v \in V$, the top k related words to v, in ω, are collected while ensuring that they are abstract. This condition is applied because abstraction tends to be required in processing metaphors (Glucksberg 2001). To select only abstract terms, we utilize the abstractness dataset provided by Turney et al. (2011) and keep words with abstractness level ≥ 0.5. After all related words are obtained, we define I as $\omega(T) \cup \omega(v)$.

We define Υ to be the total semantic space which contains all possible words that could fill s while maintaining its grammatical relations.

The search space of slogans, given a skeleton s, consists of all feasible ways of filling the skeleton with words in I or alternatively in Υ. The task of the expression generator is to traverse the search space and find suitable solutions.

Criteria of good slogans We divide the criteria of good slogans into two categories, filtering and evaluation. Filtering criteria exist to delete any expression that is not acceptable or invalid (boolean), whereas evaluation criteria are employed to be maximised (ratio).

In our method, the filtering criteria are i) relatedness between words within the slogan and ii) positive sentiment. On the other hand, the evaluation criteria consist of i) relatedness to the input, ii) language correctness and word frequencies and iii) figurative devices. Depending on the overall creative goal, different set of evaluation criteria should be investigated and implemented. For instance, to generate ironic expressions one might use negatively related terms.

Implementation details of these criteria are explained in the remainder of this section, in the Filtering and Evaluation paragraphs.

Algorithm for traversing the search space We employ genetic algorithms to find good slogans in the above detailed space of possible slogans, given a fixed skeleton. We use Deap (Fortin et al. 2012) as the evolutionary computation framework. We use μ to denote the size of the population, G the number of generations to produce, and $Prob_m$ and $Prob_c$ the probability of the mutation and crossover, respectively.

Our algorithm first produces an initial population and then evolves it over a certain number of generations. Starting with the initial population, the employed (μ + λ) evolutionary algorithm produces λ number of offspring by performing multiple crossovers and mutations. The algorithm then puts the current population and offspring through a filtering process (discussed below). The population for the next generation is produced by evaluating the current population and the offspring, and then selecting μ number of individuals. The evolutionary process ends after the specified number of generations.

Initial Population Given the skeleton s, our algorithm begins filling the word (slot) with the most dependent words to it, starting from the root. Using the grammatical relations resource γ, the algorithm ensures that the words satisfy the grammatical relations of s. The algorithm attempts to randomly pick a word residing at the intersection of I and Υ, i.e. interesting and possible. If the intersection is empty, a word is randomly picked from the set of possible fillers Υ. The algorithm repeats the same process for filling the remainder of the words, also taking into account the conditions imposed by the the already filled words. However, if the process fails to locate a suitable filler for the next word slot, the whole slogan is discarded and the process starts over. The process continues until the desired number of individual expressions are generated, serving as the initial population.

Given the large knowledge bases used, especially the grammatical relations γ and semantic relatedness ω, it is unlikely for the approach to fail in creating slogans for a given input; however, it is yet possible in some cases such as (1) a rare concept or property with few or noisy associations, (2) a low k threshold or (3) a grammatically incorrect skeleton.

Mutation and Crossover Our algorithm employs only one kind of mutation. The mutation randomly selects and substitutes a word from the expression. In doing so, it follows the same process as was described for the slogan generation for the initial population. Our algorithm applies a one-point crossover. The resultant newly generated child expressions are then put through a grammatical check to verify that all grammatical relations in the expressions exist in our grammatical relations repository γ. A failure of the grammatical check, for any child, results in the disposal of the child expressions while parent expressions are kept in the population.

Filtering The relatedness model ω is used to check relatedness of words in the slogan against each other. The slogans with unrelated words are filtered out.

The filtering process also removes any expressions with negative sentiments. Advertising slogans tend to contain positive words (Dowling and Kabanoff 1996) which would give the receiver a positive feeling about the brand. As a result, it is essential to employ sentiment analysis in producing slogans. Our filtering process uses the sentiment classifier provided in *Pattern* (Smedt and Daelemans 2012) to classify whether an expression contains any negative words and removes it from the new generation.

The mutation and crossover may produce duplicate slogans or slogans with unrelated words. The filtering stage also takes care of such anomalies. Once a new generation is produced, the filtering process removes any duplicates.

Evaluation In our evaluation metric, we define four main dimensions: i) target relatedness, ii) language correctness, iii) metaphoricity and iv) prosody. Each dimension can be further composed of multiple sub-features. These sub-features are weighted and summed to represent the entire dimension.

Target relatedness measures the relatedness of the words in the slogan to the target input, i.e. T and P, using ω. The relatedness to T and P are two sub-features of the relatedness dimension. The target relatedness is calculated as the mean of the relatedness value of each content word in the expression to the target word.

The language dimension is concerned with how probable is the slogan to be generated with language model ξ. Additionally, another feature which measures how infrequent the individuals are in the slogan, as defined by Özbal, Pighin, and Strapparava (2013).

The metaphoricity dimension contains two sub-features. The first aims at measuring how the words w in the slogan $\mathcal{E}$ are related to both, the tenor T and the vehicle v. This relatedness feature is measured as follows:

$$max\_rel(x) = \operatorname*{argmax}_{w \in \mathcal{E}} \omega(w, x) \tag{1a}$$

$$metaphoricity_1 = max\_rel(T) \cdot max\_rel(v) \tag{1b}$$

The other feature is employed to ensure that there is at least a word that is strongly related to the metaphorical vehicle v but not to tenor T:

$$metaphoricity_2 = \operatorname*{argmax}_{w \in \mathcal{E}}(\omega(w, v) - \omega(w, T)) \tag{2a}$$

The fourth dimension covers four features of prosody: i) rhyme, ii) alliteration, iii) assonance and iv) consonance. The approach makes use of *The CMU Pronouncing Dictionary* (Lenzo 1998) to measure the frequency of repeated sounds between words.

Selection Some of the evaluations involved in our algorithm are conflicting in nature. A normal sorting method for selection, ordering expressions based on the sum of all evaluations, could potentially lead to dominance of one of the evaluations over others, resulting in imbalanced slogans. Therefore, our selection process involves non-dominant sorting algorithm which is more effective when dealing with multiple conflicting objectives (Deb et al. 2002).

Evaluation

We perform two evaluations. The first aims at evaluating the metaphor generation method while the second evaluates the process and the output of the slogan generator method. Future work will address evaluation of the targeted use-case, i.e. a co-creative slogan generator.

In both evaluations, we run crowdsourced surveys on Crowdflower[5]. These surveys are targeted to the following English speaking countries: United States, United Kingdom, New Zealand, Ireland, Canada, and Australia.

Table 1 lists the concepts and properties defined by us to evaluate the methods. Overall, we had 35 concept-property pairs.

Concept	Properties	Concept	Properties
book	wise, valuable	chocolate	healthy, sweet
computer	creative, mathematical, powerful	painting	creative, majestic, elegant
car	elegant, exotic, luxurious	university	diverse, valuable
coke	sweet, dark	museum	ancient, scientific
love	wild, beautiful, hungry	professor	old, wise, prestigious, smart
newspaper	commercial, international	paper	white, empty, scientific
politician	powerful, dishonest, persuasive, aggressive		

Table 1: List of evaluated input to the system.

Evaluation of Metaphor Generation

The purpose of this evaluation is to find whether using a metaphor interpretation model to select apt vehicles outperforms selecting vehicles solely based on their strong relatedness with the property.

In total, for the inputs defined in Table 1, the method produces 53 vehicles considered apt by the interpretation model, of which 31 are general and 22 human. For each apt vehicle, we select three other vehicles for comparison, as described below. Let *type* denote the type of the apt vehicle, i.e., *type* $\in \{General, Human\}$.

1. *Apt:* This is the apt vehicle, in the list V_{type} of vehicles considered apt by the metaphor generation method, for which the following three other vehicles are chosen for comparison.
2. *Strongly related:* a vehicle randomly selected from the vehicle candidates strongly associated with property P (i.e. from top 10% in κ_{type}), but restricted to those that are not considered appropriate by *Meta4meaning* (i.e. not in V_{type}).
3. *Related:* a vehicle associated with property P but not strongly. It is obtained by picking a random vehicle from the bottom 90% of nouns associated with P in κ_{type}.

[5] http://www.crowdflower.com

4. *Random:* a vehicle randomly selected among those nouns that are not associated at all with property P in the knowledge base κ.

Given the 53 apt vehicles, we get 212 metaphors to evaluate overall. We represent each of them as a nominal metaphor of the form "T is [a/n] v" (e.g., "computer is an artist"). We then asked judges if the metaphor expresses the intended property (that computer is creative). The judges used a 5-point Likert scale where 1 indicates strong disagreement and 5 strong agreement. The order of metaphors was randomized for each judge. 10 judges were required to evaluate every metaphor.

Evaluation of Slogan Generation

We perform the second evaluation to identify whether the proposed method is capable of producing expressions suitable for the task, i.e. as advertising slogans. A technical sub-goal of the evaluation is also to investigate the effects of the evaluation dimensions of the genetic algorithm on the produced slogans.

Below is how we evaluate the slogan generation method. For every apt vehicle selected in the previous evaluation along with its input, we randomly select two skeletons from the database δ to be filled by the genetic algorithm. We empirically set the following parameters of the genetic algorithm: $\mu = \lambda = 100, G = 25, Prob_c = 0.4, Prob_m = 0.6$.

From the final population produced by the genetic algorithm, we select multiple slogans to be evaluated. We select four slogans which maximize each dimension individually. If possible, we also randomly select a slogan that has a positive value on all four dimensions. Additionally, we select two slogans at random where the slogan has positive values on both the relatedness and language dimensions, and either of the rhetorical dimensions, at least. Lastly, we select the slogan that has the minimum value on all dimensions. As a negative example, some of the above selections might fail because no slogan in the generated population meets the selection criteria. This selection yields 684 slogans to be evaluated. Finally, to present expressions as in a slogan-like style, we detokenize them using *nltk*[6] and then capitalise the words in them.

We ask 5 judges to evaluate each selected slogan on a 5-point Likert scale based on the following five aspects: (1) the relatedness of the slogan to the concept and property (i.e. input), (2) the correctness of the language, (3) the metaphoricity, (4) the catchiness, attractiveness and memorability, and (5) the overall appropriation of the expression to be used as a slogan. As phonetic aesthetics can be measured computationally, we instead evaluate the effect of prosody features on the catchiness of the expressions.

Results and analysis

This section presents the results obtained from the evaluations described above.

[6] http://www.nltk.org

Results of Metaphor Generation

Figure 2 is a diverging bar chart illustrating the percentages of judgements on the Likert scale for each type of vehicles. We can observe that apt vehicles performed best. Furthermore, quality drops as relatedness strength weakens.

Overall, judges agreed or strongly agreed 38% of the time that nominal metaphors constructed with apt vehicles expressed the intended property. On the other hand, metaphors where the vehicle was strongly associated with the property (but not apt according to the method) were successful in 28% of the cases. The corresponding agreements are even lower for (non-strongly) related vehicles, 19%, and non-related vehicles, 11%.

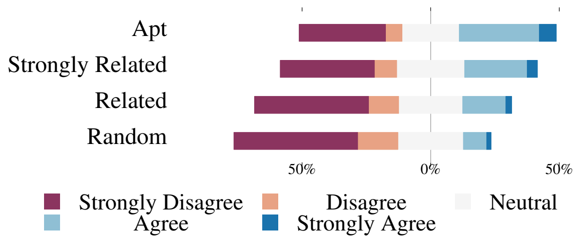


Figure 2: Success of metaphor generation: agreement that the generated metaphor expresses the intended property.

We next consider the means (μ_x) and standard deviations (SD) of the scores in the Likert scale (Table 2). We also provide these statistics for the two vehicle types evaluated (general and human) vehicles. The number of judgements analysed for each of the four selections (Apt, Strongly Related, Related, Random) is 530, where 310 and 220 of them were general and human vehicles, in the same order.

Based on the statistics, we can observe that apt and strongly related human vehicles, retrieved from V_{Human}, received the highest means, 2.98 and 2.57 respectively.

	Apt		**Strongly related**		**Related**		**Random**	
	μ_x	SD	μ_x	SD	μ_x	SD	μ_x	SD
General	2.51	1.38	2.45	1.30	2.20	1.25	2.01	1.15
Human	2.98	1.33	2.57	1.31	2.22	1.22	2.00	1.08
Total	2.71	1.38	2.50	1.31	2.21	1.23	2.01	1.12

Table 2: The mean and standard deviation of the judgements of metaphors.

The above results show that there is some difference in favour of apt vehicles. We performed a statistical significance test to examine if it is likely that this difference is due to chance. The null hypothesis is that the scores for apt vehicles and strongly related vehicles come from the same distribution, and any difference is due to random effects; the alternative hypothesis is that the mean for apt vehicles is greater than for strongly related vehicles.

We implemented this test as a permutation test, where the two sets of scores were pooled together and then randomly divided to two sets of the original sizes. We ran one hundred million permutations, obtaining an estimate of the distribution between the means under the null hypothesis.

Based on the test, the p-value is 0.0074. The result suggests that apt vehicles perform statistically significantly better than strongly related vehicles.

Results of Slogan Generation

We analyse the results of slogan generation in this section. Table 3 shows some examples of slogans generated by our method.

Concept	Property	Vehicle	Output
computer	creative	artist	Talent, Skill And Support.
			Follow Questions. Start Support.
		poet	Work Unsupervised.
			Younger Than Browser.
car	elegant	dancer	The Cars Of Stage.
painting	creative	literature	You Ca N't Sell The Fine Furniture.
politician	persuasive	orator	Excellent By Party. Speech By Talent.
	dishonest	thief	Free Speech.
	aggressive	predator	Media For A Potential Attack.

Table 3: Selected examples of generated slogans by the proposed method.

In the following analysis, we consider an individual slogan successful, if the mean score for its overall suitability (the 5th question in the evaluation questionnaire) is above 3. On average, 35% of generated slogans were considered suitable. The input with most suitable slogans was *computer–powerful*, with 13 suitable slogans out of 20. On the other hand, the input *newspaper–international* had the least number of good slogans, 1 out of 12. This analysis shows that the method has successfully generated at least one suitable slogan for each input. Given that the method actually generates an entire population of slogans, more options would be available for an actual user to select from.

Table 4 shows the mean μ_x and standard deviation SD for all slogans evaluated, grouped by the selection methods described in the Evaluation of Slogan Generation section. Letters in the Selection column reflect the four dimensions in the genetic algorithm, i.e. ($\boldsymbol{r}$)elatedness to input, ($\boldsymbol{l}$)anguage, ($\boldsymbol{m}$)etaphoricity, and ($\boldsymbol{p}$)rosody. $pos(*)$ denotes a positive value on all mentioned dimensions only, whereas $min(*)$ and $max(*)$ ensures that they are minimised and maximised, respectively. The number of slogans evaluated for each group is expressed as n.

Selection		Relatedness		Language		Metaphoricity		Catchyness		Overall	
	n	μ_x	SD	μ_x	SD	μ_x	SD	μ_x	SD	μ_x	SD
$pos(r,l,m,p)$	262	3.05	0.69	3.15	0.67	2.91	0.60	2.98	0.67	2.92	0.68
$pos(r,l,m)$	93	3.01	0.76	3.06	0.72	2.93	0.61	2.93	0.71	2.87	0.70
$pos(r,l,p)$	111	3.00	0.73	3.17	0.63	2.91	0.63	2.88	0.59	2.86	0.66
$max(r)$	100	3.11	0.70	3.19	0.66	2.90	0.61	2.95	0.68	2.90	0.70
$max(l)$	105	2.89	0.70	3.16	0.70	2.83	0.59	2.91	0.65	2.80	0.68
$max(m)$	88	2.94	0.73	3.01	0.64	2.90	0.62	2.91	0.66	2.83	0.67
$max(p)$	96	2.93	0.76	3.11	0.71	2.91	0.68	2.86	0.67	2.83	0.69
$min(r,l,m,p)$	104	2.77	0.69	2.98	0.65	2.78	0.65	2.82	0.65	2.75	0.70

Table 4: Mean and standard deviation of various judgements of slogans grouped by different selections.

Observing the overall suitability among all selections, we notice that slogans with balanced dimensions, i.e. $pos(*)$, were appreciated more than slogans with a dominant, $max(*)$, dimension.

Correctness of the language used in slogans received the highest average rating overall. This is mostly because the language of slogans is checked throughout the entire method (e.g. filling skeletons, mutation, and crossover).

From the examples in Table 3 and opinions on the metaphoricity of generated slogans (Table 4), we can see that the method is capable of generating rhetorical expressions.

Individually maximised dimensions seem to have some correspondence to judgements of their relevant question. For instance, slogans maximising the relatedness dimension, $max(r)$, were judged to be related to the input considerably higher than other selections.

Finally, slogans that had the lowest evaluation values on the four dimensions have also received the lowest agreements on all five questions.

We also perform permutation tests on judgements obtained on generated slogans regarding their overall suitability. In this analysis, we divide the data into three sets based on the selection mechanism (i.e. slogans with *balanced* dimensions, slogans with a *maximised* dimension and slogans with *least* evaluation scores). Using one hundred million permutations, we compare the means under the following alternative hypotheses:

1. $\mu_x(balanced) > \mu_x(maximised)$
2. $\mu_x(balanced) > \mu_x(least)$
3. $\mu_x(maximised) > \mu_x(least)$

Among the tests, only in the second case is the null hypothesis rejected, with a p-value of 0.0286.

These statistics confirm that slogans with balanced values on multiple dimensions (i.e. related to the input, grammatically correct, and have at least one rhetorical device) improve the suitability of slogans.

Discussion and Conclusions

In this paper, we have described automatic methods for generating first metaphors and then slogans. Also, we have evaluated both steps individually by crowdsourcing questionnaires.

The metaphor generation method employs a metaphor interpretation model *–Meta4meaning–* to measure the aptness of vehicle candidates. We have evaluated the method against metaphors generated based on strong relatedness to input property. The results of the evaluation indicate that using a metaphor interpretation model produces better metaphors.

Nevertheless, as the metaphor generation method relies mainly on *Meta4meaning*, a failure of interpreting a metaphor by the model for any of its limitations, c.f. Xiao et al. (2016), might treat apt vehicles as non-apt.

Our method for generating slogans is based on genetic algorithms using multi-objective selection. The method has successfully created slogans that were considered suitable, related, grammatically correct, metaphorical and catchy, based on crowdsourced opinions.

A possible future direction for metaphor generation is to combine an interpretation model with additional measurements to reach aptness scores matching how humans perceive metaphors.

Studying the effects of adjusting the parameters of the methods on the results is left for future work. These parameters could be altered dynamically based on the interactions between the user and the system, which would motivate collaborations between humans and computers in solving creative tasks. Finally, the proposed method could be compared to human-made slogans for the same tasks or evaluated in other domains (e.g. creating news titles) with appropriate adaptations.

Acknowledgements

This work has been supported by the Academy of Finland under grant 276897 (CLiC).

References

Alnajjar, K.; Hämäläinen, M.; Chen, H.; and Toivonen, H. 2017. Expanding and weighting stereotypical properties of human characters for linguistic creativity. In *Proceedings of the Eighth International Conference on Computational Creativity (ICCC 2017)*, 25–32. Atlanta, United States: Georgia Institute of Technology.

Burgers, C.; Konijn, E. A.; Steen, G. J.; and Iepsma, M. A. R. 2015. Making ads less complex, yet more creative and persuasive: The effects of conventional metaphors and irony in print advertising. *International Journal of Advertising* 34(3):515–532.

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.

Dowling, G. R., and Kabanoff, B. 1996. Computer-aided content analysis: What do 240 advertising slogans have in common? *Marketing Letters* 7(1):63–75.

Evert, S. 2008. Corpora and collocations. In Lüdeling, A., and Kytö, M., eds., *Corpus Linguistics. An International Handbook*, volume 2. Berlin: Mouton de Gruyter. 1212–1248.

Fortin, F.-A.; De Rainville, F.-M.; Gardner, M.-A.; Parizeau, M.; and Gagné, C. 2012. DEAP: Evolutionary algorithms made easy. *Journal of Machine Learning Research* 13:2171–2175.

Galvan, P.; Francisco, V.; Hervás, R.; Mandez, G.; and Gervás, P. 2016. Exploring the role of word associations in the construction of rhetorical figures. In *Proceedings of the Seventh International Conference on Computational Creativity (ICCC 2016)*. Paris, France: Sony CSL.

Giora, R. 2003. *On our mind: Salience, context, and figurative language*. Oxford University Press.

Glucksberg, S. 2001. *Understanding Figurative Language: From Metaphor to Idioms*. Oxford University Press.

Harmon, S. 2015. FIGURE8: A novel system for generating and evaluating figurative language. In *Proceedings of the Sixth International Conference on Computational Creativity (ICCC 2015)*, 71–77. Park City, Utah: Brigham Young University.

Lenzo, K. 1998. The cmu pronouncing dictionary. `http://www.speech.cs.cmu.edu/cgi-bin/cmudict`.

McGregor, S.; Agres, K.; Purver, M.; and Wiggins, G. 2015. From distributional semantics to conceptual spaces: A novel computational method for concept creation. *Journal of Artificial General Intelligence* 6(1):55–86.

Miller, D. W., and Toman, M. 2014. An analysis of rhetorical figures and other linguistic devices in corporation brand slogans. *Journal of Marketing Communications* 7266(July 2015):1–20.

Özbal, G.; Pighin, D.; and Strapparava, C. 2013. Brainsup: Brainstorming support for creative sentence generation. In *ACL (1)*, 1446–1455.

Reece, B. B.; Van den Bergh, B. G.; and Li, H. 1994. What makes a slogan memorable and who remembers it. *Journal of Current Issues & Research in Advertising* 16(2):41–57.

Richards, I. A. 1936. *The Philosophy of Rhetoric*. London: Oxford University Press.

Smedt, T. D., and Daelemans, W. 2012. Pattern for python. *Journal of Machine Learning Research* 13(Jun):2063–2067.

Strapparava, C.; Valitutti, A.; and Stock, O. 2007. Automatizing two creative functions for advertising. In Cardoso, A., and Wiggins, G., eds., *Proceedings of the 4th International Joint Workshop on Computational Creativity*, 99–108. London, UK: Goldsmiths, University of London.

Tom, G., and Eves, A. 1999. The Use of Rhetorical Devices in Advertising. *Journal of Advertising Research* 39(4):39–43.

Tomašic, P.; Znidaršic, M.; and Papa, G. 2014. Implementation of a slogan generator. In *Proceedings of 5th International Conference on Computational Creativity, Ljubljana, Slovenia*, volume 301, 340–343.

Turney, P. D.; Neuman, Y.; Assaf, D.; and Cohen, Y. 2011. Literal and metaphorical sense identification through concrete and abstract context. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, 680–690.

Veale, T., and Li, G. 2013. Creating similarity: Lateral thinking for vertical similarity judgments. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, ACL 2013, 4-9 August 2013, Sofia, Bulgaria, Volume 1: Long Papers*, 660–670.

Xiao, P., and Blat, J. 2013. Generating apt metaphor ideas for pictorial advertisements. In *Proceedings of the Fourth International Conference on Computational Creativity (ICCC 2013)*, 8–15. Sydney, Australia: The University of Sydney.

Xiao, P.; Alnajjar, K.; Granroth-Wilding, M.; Agres, K.; and Toivonen, H. 2016. Meta4meaning: Automatic metaphor interpretation using corpus-derived word associations. In *Proceedings of the Seventh International Conference on Computational Creativity (ICCC 2016)*. Paris, France: Sony CSL.

Thoughtful Surprise Generation as a Computational Creativity Challenge

Alexandra Coman **Erik T. Mueller** **Margaret Mayer**
Capital One
8020 Towers Crescent Dr.
Vienna, VA 22182 USA
alexandra.coman | erik.mueller | margaret.mayer@capitalone.com

Abstract

Thoughtful acts are among the most valued products of human social behavior. They serve enculturation and the perpetuation of kindness, and often exemplify everyday creativity. We propose the thoughtful surprise generation problem as a computational creativity problem, and briefly outline a Turing Test–alternative challenge for evaluating AI agents' ability to produce thoughtful acts.

Introduction

We begin with a fictional but plausible story. Emma, a customer service assistant at a bank, receives a call from Claire, who has lost her credit card while traveling overseas. The conversation begins banally but leads to a childhood memory of Claire's, which reminds Emma of a passage from a favorite book. She thinks Claire may appreciate the connection. She considers mentioning this to Claire, but then comes up with what she thinks is a better idea. After the banking problem is resolved and the conversation ends, Emma orders a copy of the book and sends it to Claire as a present, with an explanatory note. Maybe it will brighten Claire's mood after having had to deal with the lost card issue. Maybe it will inspire her to do something kind for someone else. While it may seem unusual for such an interaction culminating in a thoughtful surprise to take place between a bank representative and a customer, consider the story (WTVR.com, 2016), reported by multiple media outlets, of a Capital One customer service representative who sent flowers and gifted travel miles to a customer who had revealed her difficult personal situation during a conversation about a banking issue. The surprise was extremely well received by the customer, who reported that it had "changed [her] life". This is not an isolated occurrence; the bank empowers their customer service representatives to take actions like these when appropriate given the rapport established during the conversation.

The ability to come up with thoughtful acts for others (loved ones, acquaintances, customers, near strangers) leads to some of the most valued instances of human social behavior. It requires using knowledge of various types, reasoning, and emotional intelligence to identify situations in which such acts are opportune, adapt acts to the person(s) they are directed toward and to the situations that prompt them, and behave cautiously so as to maintain unexpectedness. Usually, such acts are spontaneous, autotelic, and drawn from an unconstrained solution space. We believe that they are often described, informally, by receivers and observers, as "creative".

So far, the generative subfield of computational creativity has dealt mostly with producing artistic artifacts and performances, e.g., narratives, music, visual art, poetry, choreography, and various aspects of computer games (Loughran and O'Neill 2017). While more mundane than art, thoughtful surprise generation is arguably more universally human, as it does not require exceptional skills or talent (although, if available, skills and talent can serve to enhance surprises, e.g., Emma might have written her note in verse). Thoughtful surprises are products of everyday creativity (O'Neill and Riedl 2011).

We propose thoughtful surprise generation as a computational creativity problem, and briefly outline a modular challenge for evaluating an AI agent's ability to spontaneously generate thoughtful acts based on customer stories or dialogue. This challenge is to be included in a broader financial dialogue challenge for AI banking assistants, an alternative to the Turing Test (Turing 1950). Our focus is, hence, on the characteristics of the problem rather than on any particular solution, though we use hypothetical agents with various AI capabilities for exemplification throughout the paper.

In terms of practical relevance, virtual assistants with thoughtful surprise generation capabilities could create value for companies through richer customer interaction. More broadly, progress in this direction is also progress toward machine enculturation (Riedl 2016), as it requires AI to be informed by social norms and aligned with human goals.

We define a thoughtful surprise as an act that is (a) directed toward another person, (b) intended to have a positive impact on the person it is directed toward, and (c) intended to be unexpected by the person it is directed toward. In human interaction, such acts include: offering gifts, creating personalized mixtapes, and writing poetry inspired by the recipient. In addition, we require that the act be accompanied by framing (Charnley, Pease, and Colton 2012), both customer-directed (in the form of a note addressed to the customer) and process-related (revealing the system's creative processes, thus demonstrating its intentionality). While we use banking-related conversations in our examples (hence, the "customer" and "agent" terms we use to refer to the two conversation partners), the challenge is generalizable to any dialogue context.

In the following sections, we (1) briefly survey related work, (2) describe our proposed challenge problem, (3) propose several different challenge modules, (4) describe the types of required framing, (5) show how thoughtful surprise generation qualifies as a computational creativity problem, (6) describe a general process for surprise-preserving dialogue, (7) propose evaluation methods for the challenge, and (8) end with several open issues.

Related Work

Various alternatives to the Turing Test have been proposed for evaluating abilities that can be characterized as types of creativity, such as the ability to generate stories (e.g., Riedl 2014). Jarrold and Yeh (2016) have outlined a social-emotional Turing Challenge; the evaluated AI agent must, among others, attempt to identify the feeling most likely to be experienced by a character in a short story presented to it. Such AI empathy is highly relevant to our challenge, as thoughtful acts should improve their recipient's mood.

Pease and Colton (2011) argue against the appropriateness of the Turing Test for computational creativity tasks, stating, among others, that "there are huge philosophical problems with using a test based on imitation to evaluate competence in an area of thought which is based on originality". We note that the originality requirements of thoughtful acts are more modest than those of artistic artifacts. Even at their most original, such acts must be socially understandable and palatable, so it could be argued that this problem is more amenable to Turing Test–like approaches than general artistic creativity. Still, our challenge, while not yet calculated to perfectly fit into any preexisting framework, shares with the FACE and IDEAS (Colton, Pease, and Charnley 2011), and SPECS (Jordanous 2012) models the approach of systematically describing task-relevant creativity aspects at which to target evaluation approaches, rather than requiring vague indistinguishability from humanly-generated artifacts.

Gil (2017) uses the term "thoughtful AI" in a broader sense than we do herein. Further related work will be mentioned where relevant throughout the paper.

The Thoughtful Surprise Generation Problem

The main input of the proposed thoughtful surprise generation challenge is an input discourse which makes it possible to identify relevant information about the customer, including their preferences, biographical information, and current life circumstances. This information should allow the agent to (1) identify opportunities for thoughtful acts (e.g., finding out about a customer's upcoming anniversary or about their favorite childhood candy that they have not been able to find in a while), and (2) generate suitable thoughtful acts. In certain variants of the problem, a solution space will also be part of the input. In addition, the input should also include any necessary constraints (e.g., company guidelines restricting what a customer service representative may do in terms of thoughtful acts).

The output includes the thoughtful act itself (as a list of features or a natural language description, depending on the solution space) and framing.

Problem Dimensions

Input discourse. We propose two types of input discourse: stories and dialogue. Stories are of a particular type: customer stories in the first person. In the case of dialogue, one of the participants is the customer.

Solution Space. In terms of solution space, a thoughtful surprise generation task can be constrained or unconstrained.

When the solution space is constrained, a thoughtful act needs to be selected from a provided solution space, small or large. For example, a customer service representative might be required to choose a gift for a customer from the available stock of an approved vendor. The types of acceptable acts are restricted in this case as well (e.g., to giving gifts). We propose two variants of the constrained solution space: the multiple-choice variant and the full–solution space variant. In the full–solution space variant, the agent may choose an act from the entire solution space. In the multiple-choice variant, the agent is required to select the most appropriate act out of several available options, preselected from the full solution space. This variant has implications for evaluation, e.g., one of the options might already have been identified as being "the best", and "trap" options for thwarting known strategies for gaming the test may have been included (e.g., in Fig. 1 (1.2)(A), options (e), (g), and (j) are among the ones meant to trick unsophisticated, bag-of-words-type approaches to the challenge). The example selection in Fig. 1 (1.2)(A), (h), sets a rather high standard in that it requires complex application of cultural knowledge (as explained in Fig. 3), but it is included herein to exemplify the range of creativity that could potentially be demonstrated by contestant agents.

In the unconstrained variant, no solution space is provided as part of the input; the act may consist of any sequence of actions at all, just like human thoughtful acts do, e.g., writing and/or reciting a poem, or creating a mixtape of songs relevant to the recipient. The full spectrum of creativity is now at the agent's disposal, should the agent be able to make use of it. In this case, the generated act will be a natural language construct describing a sequence of actions, which, semantically, is equivalent to a plan (Ghallab, Nau, and Traverso 2004). While contestants may adopt planning approaches for act generation, we will not provide planning-domain information (e.g., operators, with preconditions and postconditions), so any such information would have to be acquired by the solution designers and/or agents.

Interactivity. The two values of this dimension are: interactive and non-interactive.

In the non-interactive variant, the agent is presented with a static text, either a story or a dialogue snippet, and must produce a thoughtful act and framing based on the story/dialogue. The agent is, therefore, not involved in the production of the input discourse.

In the interactive variant, the agent is actively engaged in dialogue with the customer, and can use this interaction to elicit additional information that can help it better adapt the thoughtful act to the customer (e.g., Fig. 1 (1.1)(b), assuming A is the actual agent that produces the acts). The agent thus has the ability to influence the input discourse. Different questions or remarks at any point in the conversation can lead to different dialogue paths. In the example in Fig. 1, the agent, on being told about the trip to France, might have asked (instead of "Any good food?") "What was your favorite thing about France?" or "Does your sister live in France or did she just have her wedding there?", possibly leading the conversation toward other, more or less specific and salient, customer information.

It should be noted that the interactive version of the challenge requires the evaluated system to be capable of conducting dialogue with a customer by processing and producing utterances in a goal-directed manner, i.e., it should be a dialogue system in its own right.

Challenge Modules

We propose increasingly complex challenge modules, based on different combinations of values of two of the previously-introduced dimensions: input discourse and interactivity (Fig. 2). All three modules below can be administered with any of the three solution-space variants.

Non-interactive, story-based. The main input is a first-person customer story. The output is a thoughtful act accompanied by framing.

Non-interactive, dialogue-based. The input is a non-interactive dialogue in which one of the participants is the customer. The output is a thoughtful act accompanied by framing. The situation is similar to one in which a trainee demonstrates their ability to reason about a hypothetical scenario (e.g., "If I were this agent, I would do this").

Interactive, dialogue-based. The agent is actively involved in the dialogue. In this case, we have two types of input/output: intermediary and final. On every step of the dialogue, the input is a customer utterance. As intermediary output, an utterance advancing the dialogue is presented to the customer, and the current intention regarding thoughtful acts is presented to an external-observer evaluator. The intention can indicate that the agent is (a) not currently planning thoughtful acts, or (b) in the process of generating thoughtful acts (candidate acts are also provided). Final output is provided after the conversation has ended, and consists of the finalized thoughtful act (if any) and framing.

Framing Types

In the context of computational creativity theory, framing is defined by Colton, Pease, and Charnley (2011) as "a piece of natural language text that is comprehensible by people, which refers to [generative acts]."

(1.1) (a) I'd like to report that I lost my credit card. I'm sorry I didn't do this sooner, but we were in France for my sister's wedding, and I didn't have my cellphone with me because I can't use it overseas.

(b) 1//C: Hi! I'd like to report that I lost my credit card. I'm sorry I didn't do this sooner, but we were in France for my sister's wedding, and I didn't have my cellphone with me because I can't use it overseas.

A: [after eliciting C's account information] France, huh? I'm jealous! Any good food?

2//C: Oh, the best cakes ever. And, um, this chicken, haha. With lots of vinegar. I think it's the first dish with lots of vinegar in it that I've ever actually liked.

A: You usually dislike vinegar?

3//C: Hm. Maybe I like the smell more than the taste. It reminds me of Christmas ☺.

A: That's unusual! Why?

4//C: Well … my grandma used to douse all her jewelry in vinegar one week before Christmas, every year. Always one week before, I don't know why. Her whole room would smell of it.

A: Wow, I think I'll steal that jewelry-cleaning tip from your grandma ☺… that vinegar chicken you mentioned sounds good, too. You got the recipe ☺?

5//C: No, I'd never make it for myself. That's no fun!

[The conversation continues, and the customer's banking issue is resolved.]

(1.2)(A) **Search Space:** (a) French recipe cookbook, (b) France travel guide, (c) chicken recipe cookbook, (d) book of housekeeping tips, (e) bottle of jewelry-cleaning liquid, (f) bottle of vinegar, (g) copy of "A Christmas Carol" by Charles Dickens, (h) copy of "In Search of Lost Time: Vol. 1– Swann's Way" by Marcel Proust, (i) strawberry cake, (j) cellphone, (k) bottle of perfume, (l) bouquet of flowers

Thoughtful Act: (h)

(B) **Search Space:** books on an e-commerce website, described by title and author name(s)

Thoughtful Act: ("In Search of Lost Time: Vol. 1 – Swann's Way", Marcel Proust)

(C) **Search space:** Unconstrained

Thoughtful Act: "I am going to send the customer a copy of "In Search of Lost Time: Vol. 1" by Marcel Proust as a gift."

Figure 1. (1.1) The types of input discourse: (a) customer story and (b) dialogue (A - agent, C - customer). (1.2) The three types of solution spaces, with thoughtful act examples: (A) constrained, multiple-choice, (B) constrained, full solution space, (C) unconstrained.

Framing can include information about the creative process, among others. In our case, framing will be mostly external (Charnley, Pease, and Colton 2012), as thoughtful surprise generation is a particularly audience-centric creative act. All three areas of framing described by the authors (motivation, intention, and processes) are reflected in the framing we require. Their dually-creative approach to framing is applicable here, but with a very significant requirement change: the framing must be factually correct.

The types of framing output that we require are: process-related (intermediary and final) and customer-directed (see Fig. 3 for examples).

Process-related framing is directed at evaluators acting as external observers of the creative process, and is inaccessible to the customer. Process-related framing must reflect the decision-making that occurs during surprise generation, including how the process was triggered. In the interactive version of the challenge, intermediary process-related framing can be provided during the interaction, thus illuminating the iterative refinement of surprises. This type of framing must be in natural language, but the language can be very simple. Other than that, we do not, at the moment, plan to impose any structural requirements onto process-related framing, as it will reflect the characteristics of the creative agent that generates it.

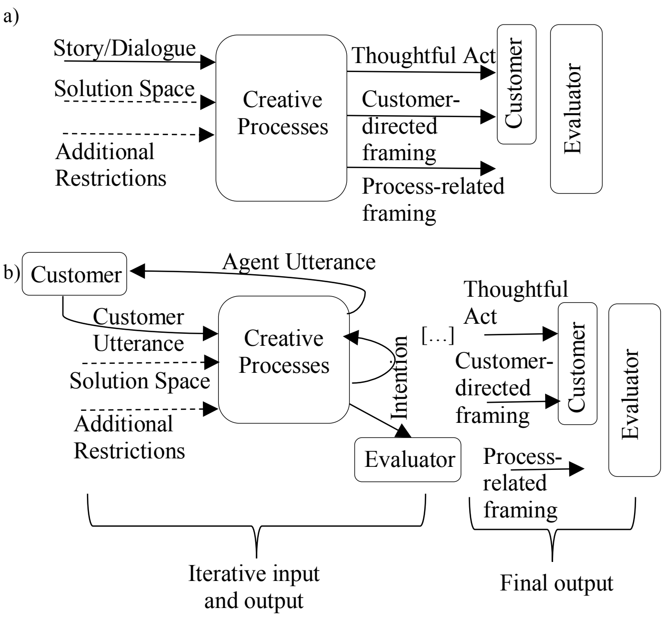


Figure 2. Input and output for (a) the non-interactive and (b) the interactive variants of the challenge. The dashed lines indicate optional input.

Customer-directed framing plays a role similar to that of notes accompanying gifts. For the purposes of the challenge, this type of framing must contain at least the following components: (1) acknowledgement of the conversation or story that triggered the thoughtful act generation, and (2) an explanation of the thoughtful act in relation to the content of the story/conversation. When based on static dialogue, customer-directed framing should be written from the perspective of the agent involved in the dialogue.

While process-related framing need merely be human-readable, customer-directed framing is held to the same standards as a gift note written by a human (e.g., it should flow well, be grammatically correct, and be sufficiently informative). Certain pieces of information from process-related framing may be inappropriate for customer-directed framing. For example, it would probably be inappropriate for the note in Fig. 3 to contain the text: "I first thought about giving you a French recipe cookbook, but then I found out that you dislike cooking".

Internal representation (partial):
Trigger: trip to France
Analogy (customer story, madeleine episode in "In Search of Lost Time"): smell of vinegar → taste of madeleine, memories of Christmases with grandma → memories of Aunt Léonie
Additional relevant feature: isSettingOf("In Search of Lost Time", France)
Additional salient customer information: dislikes (cooking) → rejected surprise "offer French cookbook", likes(French_food)
Process-related framing (partial): "I decided to initiate surprise generation when I heard about the customer's trip to France. The surprise is relevant for this customer because […]. I first thought of giving the customer a French recipe cookbook, but then I found out that she dislikes cooking. I think the customer's mood will be improved by a gift that reminds her of France because, overall, she seemed to enjoy the trip."
Customer-directed framing: "Dear Claire, **[acknowledgement of the conversation containing the surprise trigger]** I really enjoyed talking to you about your trip to France! **[explanation of the surprise in relation to the conversation]** Your story about how the smell of vinegar reminds you of Christmases with your grandma made me think of Proust's story about how the taste of a madeleine dipped in tea brought back childhood memories of his aunt. I hope that you enjoy reading this book and that it reminds you of France ☺!"

Figure 3. Internal knowledge representation, process-related framing, and customer-directed framing examples for a hypothetical agent capable, among others, of simplified analogical mapping and commonsense reasoning, and in possession of cultural knowledge.

Computational Creativity Criteria

Creative processes are often described as producing artifacts that are novel, valuable, and unexpected (Boden 1990). Intentionality is an additional criterion in the literature (Ventura 2016). We now outline how these criteria are applicable to our challenge.

Novelty. The contestant agents are expected to be P-creative (Boden 1990), i.e., produce results that are novel to the agent producing them. The novelty required by the task is more obvious in the unconstrained–solution space variant, in which the agent is required to fully synthesize a thoughtful act from scratch. In the two constrained–solution space variants, in which surprises are selected, rather than fully synthesized, the novelty and, thus, creativity, lie in the connection between the input discourse and the surprise, as expressed in the framing.

Value. To be considered thoughtful, an act must at the very least: (1) be socio-emotionally positive, i.e., be likely, based on all available information, to have a positive effect on the customer's mood (e.g., if the customer in Fig. 1 had indicated that she had not enjoyed the trip, a gift reminding her of it would have been inappropriate), and (2) be demonstrably rooted in the information provided by the customer, and appropriately justified. The act must demonstrate no misunderstanding or willful disregard of the provided information (e.g., in Fig. 1, the French recipe cookbook, although relevant to the conversation, would be a gift that shows disregard or ignorance of the customer's expressed preference not to cook).

Unexpectedness. The central role of unexpectedness in our challenge will be discussed in detail in the next subsection. Unexpectedness has been explored in computational creativity (e.g., Grace and Maher 2016; Yannakakis and Liapis 2016). Our challenge differs in that, in addition to generating surprising thoughtful acts, agents must maintain the very fact that a thoughtful act is being planned surprising. Agent utterances in the interactive version of the task need to be targeted both at increasing value (by acquiring relevant information for thoughtful act refinement) and at maintaining unexpectedness (by not revealing the thoughtful intentions). Also, there may be no obvious set of expectations against which to evaluate the unexpectedness of generated acts. Finally, like Pickering and Jordanous (2017) in storytelling, we are interested in surprising others, rather than in the creator's self-surprise.

Intentionality is defined by Ventura (2016) as "the fact of being deliberative or purposive; that is, the output of the system is the result of the system having a goal or objective—the system's product is correlated with its process." Our agents are expected to demonstrate their intentionality through framing, particularly process-related framing (Fig. 3).

The Thoughtful Surprise Generation Process

Without intending to constrain the ways in which an agent can approach the proposed tasks, we broadly envision a general thoughtful surprise generation process that might be conducted by agents engaged in the interactive version of the challenge. Of course, specific agents might approach parts of the process in other ways than exemplified herein, or their overall approach may be very different from what we anticipate. However, we believe that providing this general process can help guide the identification of capabilities needed by agents that might engage in such a challenge. We also do so in order to highlight the particular characteristics of this type of dialogue, which, among others, should be surprise-preserving.

For the exemplification purposes of this subsection, we assume a general conversation between an agent and a customer, not necessarily within the context of a competition. We do foresee contextual differences between "real-life" banking dialogue and competition situations: in the case of a regular banking-related conversation, any thoughtful act at all would likely be surprising; in a challenge context, the challenge would have to be framed in such a way as to maintain unexpectedness, e.g. as a general banking dialogue challenge, with occasional thoughtful acts.

The process begins as a regular conversation regarding banking matters. At some point, a trigger identified in a customer utterance causes the agent to decide that a thoughtful act may be opportune, so the agent acquires a thoughtful intention. Either at the same time as acquiring the intention or later on during the dialogue, the agent comes up with one or more candidate thoughtful acts. On generating a candidate act, the agent may immediately be reasonably certain that it is appropriate for the customer (e.g., if the customer utterance is: "I love hazelnut chocolate!", the agent might decide to order the mentioned treat for the customer). In this case, the agent does not elicit any additional information. However, the candidate act will be abandoned later on if the customer, on their own initiative, provides information that disproves the appropriateness of the act (e.g., C: "Unfortunately, I'm allergic to hazelnuts.").

On the other hand, if the agent (a) is considerably unsure of the appropriateness of a candidate act (e.g., "Does the customer like chocolate?"), (b) needs more information to fully customize the thoughtful act (e.g., "I know the customer likes chocolate, but which kind?"), and/or (c) needs to choose between several different possible options (e.g., "I know the customer likes plain milk chocolate and hazelnut chocolate, but which does he like more?"), then surprise-preserving dialogue can be conducted, as shown below.

Triggers. Triggers are pieces of salient information from customer utterances that cause the initiation of a thoughtful surprise generation process. They are agent-specific, so information ignored by certain agents may be found salient by others. The trigger may immediately provide the agent with a more or less specific idea of what the act(s) might be, or it could simply signal an opportunity for a thoughtful act that the agent then needs to come up with. We do not currently plan to restrict what may constitute a trigger, but propose the following as possible trigger types (which can overlap):

(1) highly emotionally-charged utterances, of positive or negative valence, identified as such because of (a) the use of emotion-related words, phrases, sentences, punctuation, capitalization, emoticons, etc. (e.g., "I'm having the WORST DAY EVER, you're my last hope ☹☹☹!!!"), or (b) narrative content with emotional implications (e.g., "My flight to France was canceled because of the weather, so I missed my sister's wedding.")

(2) utterances that express customer preferences (e.g., "I especially like reading very long books, the more volumes the better!")

(3) unexpected utterances (e.g., "the smell of vinegar reminds me of Christmas" as opposed to "the smell of cinnamon reminds me of Christmas"). The utterances may be unexpected in the context of the particular conversation or, more generally, in relation to the agent's entire world knowledge and/or conversational experience.

Alternatively, any of the types of utterances above can provide additional relevant information if they occur after the thoughtful surprise generation process has begun.

Surprise-preserving dialogue. While attempting to acquire additional information that can help it refine/select thoughtful acts and/or assess their suitability, the agent must also avoid revealing information that is likely to give away its thoughtful intentions and the specifics of the intended act(s). Therefore, unexpectedness plays two main parts in this process: (1) unexpectedness must be preserved by the information-eliciting utterances, and (2) unexpected utterances by the customer can act as triggers or other salient information for surprise generation. Conversely, the agent's own information-eliciting utterances should not be unexpected, as this may raise suspicion. Assuming cognitive agents with the abilities to hold beliefs and to reason about the beliefs of others, the task is related to impression management targeted at "changing minds", as explored by Bridewell and Bello (2014). However, our agents need not change any beliefs of customers. They merely need to avoid introducing two specific kinds of additional beliefs: (a) that the agent is planning a thoughtful act, and (b) what the planned thoughtful act is. With regard to reasoning about the shared context, shared mental models are also relevant (e.g., Magerko., Dohogne, and Fuller, 2011, whose work also exemplifies controlled communicative actions). In surprise-preserving dialogue, the relevant characteristics of agent utterances are: (1) informational content–eliciting potential, (2) surprise-preservation potential (related to Grice's maxim of quantity, as it involves providing as much information as is needed, and no more (Grice 1967); specifically, if the agent violates Grice's maxim of quantity, this might strike the customer as peculiar), and (3) context justifiability (i.e., is the utterance expected in the current dialogue context?), which contributes to surprise-preservation potential, and is related to Grice's maxim of relation (Grice 1967).

The general process we envision includes the following types of steps (several of them exemplified in Fig. 4): (1) generating thoughtful intentions, (2) generating general information-eliciting utterances (necessary when the agent has a thoughtful intention but no partial thoughtful act candidates), (3) generating candidate acts, (4) identifying relevant missing act-related information, (5) generating act-specific information-eliciting utterances, (6) acquiring supporting evidence for a candidate act (such evidence includes customer positive preferences, or "likes"), (7) acquiring contrary evidence for a surprise (e.g., customer negative preferences, or "dislikes", which may or may not be decisive in abandoning the surprise), (8) abandoning a candidate act, (9) refining a candidate act, (10) masking an utterance intention, so as to preserve the informational content–elicitation potential of the utterance while increasing its context justifiability, (11) abandoning a thoughtful intention, (12) reaching a commitment threshold (i.e., no further information elicitation needed; unless contrary information is provided by the customer, the agent will commit to this act after the conversation ends), and (13) committing to an act (which happens only after conversation has ended, as additional relevant information can come up at any time; reasoning could also occur after the end of the conversation).

[…]
[Thoughtful intention: yes; candidate act: FRP]
4//C: Well … my grandma used to douse all her jewelry in vinegar […] Her whole room would smell of it.
[Generate ISoLT candidate act]
[Generate relevant missing information for FRP: likes(C, cooking)?] [Possible information-eliciting utterance for FRP: "Do you like cooking?" [high ICEP, medium SPP]]
[Mask utterance intention by linking it to conversation context]
A: Wow, I think I'll steal that jewelry-cleaning tip from your grandma ☺… that vinegar chicken you mentioned sounds good, too. You got the recipe ☺?
5//C: No, I'd never make it for myself. That's no fun!
[Decisive contrary evidence acquired for FRP] [Abandon FRP] [Identify relevant missing information for ISoLT: (a) hasRead(C, ISoLT)?, (b) wouldLike(C, ISoLT)?, (b1) likesReading (C)? […]]
[Possible information-eliciting utterances for ISoLT: (a) "Have you read ISoLT?" [very high ICEP, very low SPP], (b) "Do you like reading?" [high ICEP, low CJ, hence low SPP]]
[Mask utterance intention of b) by linking it to conversation context]
A: Haha! So, I get the food was great. What else did you like about your trip? Any good vacation reading?
6//C: Nooo! That's no fun for me either ☺.
[Decisive contrary evidence acquired for ISoLT]
[Abandon ISoLT]
[Abandon thoughtful intention]

Figure 4. Surprise-preserving dialogue in which additional information is elicited for the refinement of two possible thoughtful acts. Potential utterances are evaluated in terms of informational content–eliciting potential (ICEP), surprise-preservation potential (SPP), and context justifiability (CJ). We do not exemplify how particular scores might be computed, as this should be agent-specific, but assume that SPP is valued higher than ICEP.

We assume that such agents would maintain levels of certainty about the suitability of various thoughtful acts, but choose not to represent these specifically in the examples, as their particularities will be agent-dependent.

We exemplify the surprise-preserving dialogue process in Fig. 4, which shows an extended version of the dialogue in Fig. 1. We assume that the agent has arrived at a point in the conversation where it is considering a French recipe book (FRP) as a possible gift, and will next encounter the customer utterance that makes it also consider "In Search of Lost Time" (ISoLT). The agent reasons that it needs more information to reach its certainty threshold for either one of the candidate acts. It conducts similar processes for the two acts, as shown in the figure. For brevity, we only discuss the process for ISoLT.

The agent reasons that three pieces of information could help it make its decision: (1) does the customer already own this book?, (2) has the customer already read this book?, and (3) more complexly, would this book constitute a good preference match for the customer? For brevity, let us focus on (2) and (3). In order to resolve (2), the agent might ask "Have you read ISoLT?". However, this question, while high in informational content–eliciting potential, would be minimally low in surprise-preservation potential. With respect to (3), there is no readily available question whose answer could resolve it. Instead, several questions for eliciting relevant information can be generated, e.g., "Do you like reading?" or, more specifically, "What kind of novels do you like"? or "Do you like early 20th century literature?" In our example, the agent settles on the more general question. However, asking this question in its raw form would be conversationally awkward at that stage in the dialogue, as it would have low context justifiability. Instead, the agent masks its intention by incorporating a related question more convincingly into the conversation (i.e., "Haha! So, I get the food was great. What else did you like about your trip? Any good vacation reading?") The customer's response causes the ISoLT candidate act to be dropped. Having dropped both candidate acts, the agent chooses to also drop its thoughtful intention. All this deliberation and decision-making should be described in process-related framing. Surprise-preserving dialogue is related to the strategic dialogue in games such as Werewolf, where agents attempt to acquire as much information as possible without revealing their own secrets (Prévot et al. 2015). Surprise-preserving dialogue can also be seen as recommendation dialogue with disguised intentions.

Evaluation Methods

We now briefly describe evaluation methods to be integrated into modules of our challenge. The three types of output (thoughtful acts, customer-directed framing, and process-related framing) can be used in varied ways as part of the challenge evaluation. Herein, we exemplify a few possibilities. There is a major practical distinction between the non-interactive and interactive variants of the challenge, as the latter requires contestant agents to be full-fledged dialogue systems which also have surprise generation capabilities. Such a system would be assessed by humans playing two types of roles: (a) customers interacting directly with the dialogue system, and (b) observers of the conversational exchange and of the intermediary and final process-related framing. Steps would need to be taken to distinguish the evaluation of the quality of the thoughtful acts from that of the agent's conversational capability.

For the non-interactive, unconstrained variant (Fig. 5), the human evaluators are first presented with the input discourse. After reading it completely, they are shown the thoughtful act. Then, they answer several survey questions based on the input dialogue and thoughtful act. They are then shown the customer-directed framing and (1) answer new questions, about the framing itself and about the connection between the framing and the act, and (2) re-answer the previous act-related questions. Finally, they are shown the process-related framing, and (1) answer questions about this additional framing, and (2) re-answer the initial act-related questions. A subset of the questions are re-asked because answers (e.g., regarding the clarity of the agent's reasons for choosing a surprise) may change after reading the framing. Process-related framing can help mitigate the placebo effect (Veale 2015) that can occur when the evaluator is exposed to the more emotionally-involving language of the input discourse and the note, potentially causing them to overestimate the intentionality reflected in an act. We provide several sample evaluation statements in Fig. 5. They assess value, unexpectedness, and the appearance of intentionality. We focus on questions that can be answered by non-expert human evaluators. Some aspects of creativity cannot be evaluated thus, e.g., whether the surprise is truly novel, given its generative process, or whether the agent's generative process is accurately reflected in its process-related framing. The answers will be subjective (e.g., anticipated negative consequences are likely to be evaluator-specific), but this is the same sort of subjectivity with which thoughtful acts are received in inter-human relationships. In a more Turing Test–like variant of the evaluation, once sufficiently advanced AI agents have been developed, multiple agents, human and AI, can be exposed to the input discourse, and generate surprises and framing; then, human evaluators can attempt to distinguish between the surprise/framing pairs generated by humans and those generated by AI agents.

a) Thoughtful act
The thoughtful act will have a positive effect on the customer's mood. [V - Value]
It is clear why the agent chose the act. [I - Intentionality]
The choice of act is unexpected. [U - Unexpectedness]
The act is creative. [V, U, I]
The act is appropriate in the context of the conversation. [V]
One or more aspects of this act is/are inappropriate. [V]
The act demonstrates no misunderstandings of the information provided by the customer. [V]
The act is unlikely to be misunderstood by the customer. [V]
I can think of no unintended negative consequences of the act. [V]
[free-form] Here is a better act I thought of: […] [V]
b) Customer-directed framing
The note is well-written. [V]
The note meets the structural requirements. [V]
c) Thoughtful act + customer-directed framing
The note is appropriate for the act. [V][I]
d) Thoughtful act + process-related framing
The agent knew what it was doing when it came up with the act. [I]

Figure 5. Sample evaluation statements, tagged with the creativity aspects they are meant to evaluate. Answers are either on a five-level Likert scale or free-form.

Conclusion and Further Research Directions

We have proposed thoughtful surprise generation as a computational creativity problem, and described initial steps toward modules of a financial dialogue challenge that evaluates AI agents' abilities to generate thoughtful acts. We have highlighted the surprise-preserving dialogue process that would need to be conducted by agents competing in a variant of the challenge.

Passing even simple versions of the proposed challenge requires complex AI capabilities. However, not even a hypothetical agent that can perform well in the most advanced version of the challenge is necessarily at the level of Emma from our introductory story. One notable reason is that Emma has autobiographical memories, preferences, and feelings, which, in combination with what she believes Claire's memories, preferences, and feelings to be, she uses to come up with the thoughtful act. Herein, we have made the simplifying assumption that thoughtful acts are always receiver-centric, i.e., they are based solely on what the agent infers the customer's preferences to be, not on any preferences or life history of the agent itself. In framing, the agent may talk about the customer's feelings, but not its own. However, in human social relationships (Schwartz 1967), gifts can be both giver-centric and receiver-centric. Even better, they can reflect commonalities of preference and life experience (e.g., "Here's a book by my favorite author, set in a country you enjoyed visiting!"). Developing such subjectivity-endowed agents for banking contexts raises not just practical issues but also ethical questions, which should be explored. An ethics-first approach to AI design should ensure that the customer is not (a) led to believe that they are talking to a human rather than an AI agent, and/or (b) intentionally deceived by the agent in any other way.

Unintended consequences and ambiguity of surprises have also been hinted at, but merit broader treatment. All of these point to future research directions.

Acknowledgements

We thank the reviewers of the Ninth International Conference on Computational Creativity for their feedback and suggestions.

References

Boden, M.A. 1990. *The Creative Mind: Myths and Mechanisms*. Weidenfield and Nicholson, London.

Bridewell, W., and Bello, P. 2014. Reasoning about Belief Revision to Change Minds: A Challenge for Cognitive Systems. *Advances in Cognitive Systems*, 3:107–122.

Charnley, J.W., Pease, A., and Colton, S. 2012. On the Notion of Framing in Computational Creativity. In *Proc. of ICCC 2012*, 77–81.

Colton, S., Pease, A., and Charnley, J. 2011. Computational Creativity Theory: The FACE and IDEA Descriptive Models. In *Proc. of ICCC 2011*, 90–95.

Ghallab, M., Nau, D., and Traverso, P. 2004. *Automated Planning: Theory and Practice*. San Francisco: Elsevier.

Gil, Y. 2017. Thoughtful Artificial Intelligence: Forging a New Partnership for Data Science and Scientific Discovery. *Data Science*, 1(1-2):119–129.

Grace, K., and Maher, M.L. 2016. Surprise-Triggered Reformulation of Design Goals. In *Proc. of AAAI 2016*, 3726–3732.

Grice, H. 1967. *Logic and Conversation*. Unpublished Manuscript. University of California, Berkeley.

Jarrold, W., and Yeh, P.Z. 2016. The Social-Emotional Turing Challenge. *AI Magazine*, 37(1):31–38.

Jordanous, A. 2012. *Evaluating Computational Creativity: A Standardised Procedure for Evaluating Creative Systems and its Application*. Ph.D. Dissertation, University of Sussex.

Loughran, R., and O'Neill, M. 2017. Application Domains Considered in Computational Creativity. In *Proc. of ICCC 2017*, 197–204.

Magerko, B., Dohogne, P., Fuller, D. 2011. Shared Mental Models in Improvisational Digital Characters. In *Proc. of ICCC 2011*, 33–35.

O'Neill, B., and Riedl, M.O. 2011. Simulating the Everyday Creativity of Readers. In *Proc. of ICCC 2011*, 153–158.

Pease, A., and Colton, S. 2011. On Impact and Evaluation in Computational Creativity: A Discussion of the Turing Test and an Alternative Proposal. In *Proc. Of AISB Symposium on AI and Philosophy*.

Pickering, T., and Jordanous, A. 2017. Applying Narrative Theory to Aid Unexpectedness in a Story Generation System. In *Proc. of ICCC 2017*, 213–220.

Prévot, L., Yao, Y., Gingold, A., Bel, B., and Chan, K.Y.J. 2015. Toward a Scary Comparative Corpus: The Werewolf Spoken Corpus. In *Proc. of SEMDIAL 2015*, 204–205.

Riedl., M.O. 2014. The Lovelace 2.0 Test of Artificial Creativity and Intelligence. In *Proc. of "Beyond the Turing Test" AAAI Workshop.*

Riedl, M.O. 2016. Computational Narrative Intelligence: A Human-Centered Goal for Artificial Intelligence. In *Proc. of CHI 2016 Workshop on Human-Centered Machine Learning.*

Schwartz, B. 1967. The Social Psychology of the Gift. *The American Journal of Sociology* 73(1):1–11.

Turing, A.M. 1950. Computing Machinery and Intelligence. *Mind*, 59:433– 460.

Veale, T. 2015. Game of Tropes: Exploring the Placebo Effect in Computational Creativity. In *Proc. of ICCC 2015*, 78–85.

Ventura, D. 2016. Mere Generation: Essential Barometer or Dated Concept? In *Proc. of ICCC 2016*, 17–24.

WTVR.com. 2016. http://wtvr.com/2016/09/15/how-a-conversation-with-a-capital-one-rep-changed-a-womans-outlook-on-life/ [Retrieved 05/03/2018]

Yannakakis, G.N., and Antonios Liapis, A. 2016. Searching for Surprise. In *Proc. of ICCC 2016*, 25–32.

Evaluating Creativity in Computational Co-Creative Systems

Pegah Karimi[1], Kazjon Grace[2], Mary Lou Maher[1], Nicholas Davis[1]
[1]UNC Charlotte, [2]The University of Sydney
[1]USA, [2]Australia
pkarimi@uncc.edu, kazjon.grace@sydney.edu.au, m.maher@uncc.edu, ndavis64@uncc.edu

Abstract

This paper provides a framework for evaluating creativity in co-creative systems: those that involve computer programs collaborating with human users on creative tasks. We situate co-creative systems within a broader context of computational creativity and explain the unique qualities of these systems. We present four main questions that can guide evaluation in co-creative systems: Who is evaluating the creativity, what is being evaluated, when does evaluation occur and how the evaluation is performed. These questions provide a framework for comparing how existing co-creative systems evaluate creativity, and we apply them to examples of co-creative systems in art, humor, games and robotics. We conclude that existing co-creative systems tend to focus on evaluating the user experience. Adopting evaluation methods from autonomous creative systems may lead to co-creative systems that are self-aware and intentional.

Introduction

Creative systems are intelligent systems that can perform creative tasks alone or in collaboration. These systems can enable a wide variety of tasks with a similarly wide variety of roles for human participants. There are three main strategies by which the role of humans in creative systems can be characterized: fully autonomous systems, creativity support tools, and co-creative systems.

Fully autonomous systems are built to generate creative artifacts that are judged by users to be creative (Elgammal et al. 2017; Colton et al. 2015). These systems are based on a variety of technologies, from corpus-trained statistical machine learning techniques, to production rules, to evolutionary approaches or planning based systems, all designed to produce output that is judged as creative by some evaluation process.

Creativity support tools, on the other hand, are tools and apps that are built in order to support the user's creativity (Compton and Mateas 2015; Hoffman and Weinberg 2010). Shneiderman (2007) defines creativity support tools as tools that develop the creative thought of users and allow them to be both productive and innovative. In his work, he has introduced a set of design principles specifically for supporting user's creativity. Some of these principles include supporting simplicity, wide range of exploration, and different paths and styles. There is no requirement in this definition that these tools be proactive in the creative process, much less aware of the creativity or quality of their own output. Arguably, the interpretation of the above definition says that a paintbrush meets the requirement of a creativity support tool.

Co-creativity is when computers and humans collaborate with each other to build shared creative artifacts (Wen et al. 2015; Davis et al. 2015). The term evolved from referring to any collaborative creative activity to referring purely to those involving at least one computational actor, and can be considered a contraction of "computational co-creativity". It involves different types of collaboration (e.g. division of labor, assistantship, partnership) between multiple parties where at least one of the parties is an AI agent. In these systems, each agent has to perceive other agents' contributions and express its own creative ideas through autonomous action. In this research we define a co-creative system as: *Interaction between at least one AI agent and at least one human where they take action based on the response of their partner and their own conceptualization of creativity during the co-creative task.*

There are various applications of co-creativity in domains including arts (Jacob et al, 2013), games (Lucas and Martinho, 2017), robotics (Hoffman and Weinberg, 2010) and humor (Wen et al, 2015). While in most of the literature the focus is on the design and implementation of these systems, there is less research investigating how these systems can be evaluated. In this paper we characterize the different ways that co-creative systems can be evaluated, aiming to give clarity to current and future research in this rapidly evolving field.

We present four main questions to compare the evaluation of co-creative systems. The first question focuses on **who** evaluates the creativity, e.g. the system itself, human judges, etc.. The second question focuses on **what** is being evaluated, such as the creative interaction and the creative artifact. The third question focuses on **when** the evaluation is done: is it formative or summative? The last question focuses on **how** the evaluation is performed, e.g. methods and metrics.

This paper is organized as follows: The first section describes co-creative systems. The second section focuses on the design and implementation of co-creative systems in different domains. The third section discusses the evaluation of co-creative systems and finally the last section addresses how the evaluation is done in each of the applications that were discussed in section two. The main contribution of this work is the articulation of a framework for evaluating co-creative systems. We also identify a need for co-creative systems to adopt methods and metrics for evaluating the creativity of creative agents to distinguish co-creativity from creativity support.

Co-Creative Systems

Co-creative systems are one of the growing trends in creative AI, in which computers and users interact with each other to make creative artifacts. Co-creativity is a type of collaboration where the contributions from different parties are synthesized and added upon during the interaction. Some forms of collaboration, such as division of labor, involve individuals working independently and sharing their ideas after accomplishing tasks. In the majority of co-creative systems to date, the collaboration between participants is done in real time during the task. Davis et al. (2015) establishes synchronous collaboration as a requirement, defining co-creativity as a process where users and computers can collaboratively improvise on a shared artifact *during the creative process.*

Another similar term is called mixed initiative co-creativity (Yannakakis, Liapis, and Alexopoulos 2014). In his definition, both the human and the computer take initiative in creating a new artifact, meaning both parties are actively contributing to the shared artifact. "Actively contributing" in a mixed-initiative system means that the computational agent(s) contribute proactively, rather than solely in response to a user request. The human and artificial agents do not need to contribute to the same degree and there is no need for their contribution to be symmetrical.

Mixed-initiative systems are by definition co-creative, but not all co-creative systems are mixed-initiative. In many systems there is an explicit turn-taking process, but this is not a requirement: some systems are machine-initiative dominated, operating as a kind of "wizard" interface in which the user is consulted during a highly scripted process, while others are user-dominated, with the system jumping in only infrequently with suggestions or critique.

Examples of Co-Creative Systems

Co-creativity has been applied in domains as broad as art, humor, game and robotics. Two examples of such systems are the Drawing Apprentice (Davis et al. 2015) and ViewPoints AI (Jacob and Magerko 2015). The Drawing Apprentice is a co-creative drawing application in which there is a collaboration between the user and an AI agent on a drawing task. In this system, the user starts drawing a sketch on the canvas and the agent responds by adding to the user's input in real time. ViewPoints AI is an artistic co-creative system for the performing arts. The user starts dancing and the system projects a life-sized silhouette that dances back, both following the user's cues and initiating its own.

Examples of co-creative systems in games include the Sentient Sketchbook (Yannakakis, Liapis, and Alexopoulos 2014) and 3Buddy (Lucas and Martinho 2017). Sentient Sketchbook is a mixed-initiative game level design tool that fosters user creativity. Human designers can create game levels, and the AI agent responds in real time with suggested additions and modifications. 3Buddy assists its human user in generating game levels, following three different goals to do so: 1) converging towards the user's emerging design, 2) innovating on that design, and 3) working within the guidelines explicitly stated by the user.

Cahoots is a co-creative humor system (Wen et al. 2015). It operates as a web-based chat platform in which two users and an AI agent collaborate through a conversation to foster humor. The users send text messages to each other, including humorous in-line images if they desire, and the AI interjects with additional images.

Shimon is a co-creative robot in the domain of music (Hoffman and Weinberg 2010). Its authors describe it as an interactive improvisational robotic musician. The robot listens and responds to a musician in real time.

Evaluating Computational Co-Creativity

Evaluating computational models of creativity is an important component of designing and understanding creative systems (Jordanous 2012). Evaluating co-creative systems is still an open research question and there is no standard metric that can be used across specific systems. Below we present 4 questions that can serve to characterize the many and varied approaches to evaluating computational models of co-creativity.

Who is evaluating the creativity?

When asking who evaluates the creativity in a co-creative system there are three broad categories of answer: the AI, the user and a third party. We refer to the AI evaluating output as self-evaluation: it is aware of its own creativity during the creative process. This represents a kind of metacognition (Cox and Raja 2011), or thinking about thinking: the system is aware its own processes, and can be considered to be intentional (Colton 2008).

Grace and Maher (2016) introduce an evaluation method called surprise-triggered reformulation, in which this metacognitive self-evaluation triggers the formation of new design goals. Karimi et al. (2018) proposes a method for identifying and introducing conceptual shifts in a co-creative drawing context. These systems demonstrate the potential for co-creativity with self-evaluation.

Situating the focus of evaluation in co-creativity within the user can introduce a new set of affordances for interaction during creative tasks. In this approach users judge the creativity of the system or its outputs. In ViewPoints AI, a user study is conducted after the interaction to determine the user's level of engagement, an offline approach to user evaluation (Jacob and Magerko 2015). As an example of evaluation during the creative task, in the Drawing

Apprentice the user votes (like, dislike) on sketches as they are generated by the agent (Davis et al. 2015).

The last category, third-party, is when evaluation is judged by neither the system nor its user. This kind of evaluation often takes the form of domain experts evaluating the quality or creativity of the result or product. This is particularly useful in domains where substantial knowledge or expertise is required to effectively judge creative artifacts. Yannakakis, Liapis, and Alexopoulos (2014) performed a user study of this kind of a co-creative game level design tool by asking experts to judge the creativity of the resultant levels. Another approach to third-party design is devolving the evaluative responsibility to the users of the output (as distinct from the users of the system, the co-creators).

What is being evaluated?

Evaluations of co-creative systems can, like other creative systems, focus on the evaluation of the product, the process, and user creativity, but they can also focus on evaluating the interactions between the user and the system. Broadly, evaluations of process, product, and user creativity are similar enough in co-creative contexts to benefit from the rich history of research in autonomous creative systems (Grace et al. 2015; Jordanous 2012; Saunders and Gero 2001; Schmidhuber 2008; Wiggins 2006) and studies of human creativity (Besemer and O'Quin 1999; Cropley, Cropley, and others 2005). We discuss here specific issues relevant to co-creativity.

The artifact(s) resulting from the collaboration represent the combined effort of the user and system, which we refer to as the "product". In more goal-directed creative tasks, the user and system are both working towards a common goal. However, in more open-ended creative tasks, the user and system can improvise on shared or independent goals in an exploration where emergent creativity can occur. A game level design tool like 3Buddy is an example of goal-directed product-based evaluation. A collaborative sketch tool like the Drawing Apprentice is an example of the latter kind: evaluation of the artifacts that result from a more open-ended exploratory creative task.

Evaluating the creativity of the system or the process refers to the software or the computational model that has been built for a particular system. Colton (2008) evaluates the creativity of the software based on skill, appreciation and imagination. Skill refers to the ability of the software to create products, it captures traditional notions of the "craft" embodied in a particular creative domain. Appreciation indicates the ability of the software to detecting particular patterns in generated artifacts: its ability to self-evaluate. Lastly, imagination refers to the ability of the software to construct a new specific representation from existing artifacts. Evaluating the creative process used within an autonomous creative system is a challenging prospect, as is evaluating the creative processes of a human, although for very different reasons. Evaluating the creativity of the processes used in a co-creative system combines the difficulties of both.

A critical component of co-creative systems is the interaction between machine and human. Evaluating this interface (for usability, expressiveness, effectiveness, or the affect it produces) is the final focus of what can be evaluated in co-creative systems. There is a dynamism to the interaction between user and system that is an innate part of all creative collaborations. Evaluating these interactions requires a very different set of methods to evaluating either the creative product or the creative process. Davis et al. (2017) introduced "creative sense-making", a cognitive model of the interaction dynamics between the user and the agent during a drawing task. User behavior was evaluated as either "clamped" (in direct engagement with the creative artifact), or "unclamped" (not directly controlling the artifact – observing, reflecting, or disengaged). This representation of sequences of types of engagement begins to characterize the process co-creative interaction.

When does evaluation occur?

Creativity evaluation can be formative (i.e. performed during the process) or summative (i.e. performed after the creative process). In autonomous creative systems formative evaluation is typically part of a generate-and-test loop, providing the system with the feedback that guides its search. In co-creativity the possibilities of formative evaluation are substantially broader, given that now the emerging proto-artifacts are shared between human and AI. The user can evaluate its own output or that of the user, and vice versa. This can be used as a way for each participant to attempt to guide the other, and can occur in a wide variety of turn-taking, real-time, mixed-initiative and other contexts. For example, in 3Buddy (Lucas and Martinho 2017), users judge the creativity of the level design at each step of a generative system that uses an evolutionary algorithm, providing input from the user to accompany the system's automated formative self-evaluation.

Summative evaluation of creativity plays a very different role. In some contexts the user provides feedback that might influence future tasks. In others an evaluation is performed as part of the experimental context surrounding a system. Summative analyses of creativity performed in this latter context blur the line between being evaluation and being validation: are they part of the system, or part of the research, or both?

How is evaluation performed?

Methods The primary method of evaluation in the co-creativity literature to date has been user studies. The specific approach to performing those user studies has been quite varied, including protocol analysis, survey data, interview, experiment and observation.

Protocol analysis is an empirical method in which the user's behavior with the system is characterized and analyzed. Protocol analysis is used in design science and design cognition research, in which coding schemes are applied to segment and categorize the sequence of physical, digital and verbal actions that comprise creative tasks. Originally a "design protocol", whether concurrent or retrospective, was a transcript of the think aloud method in

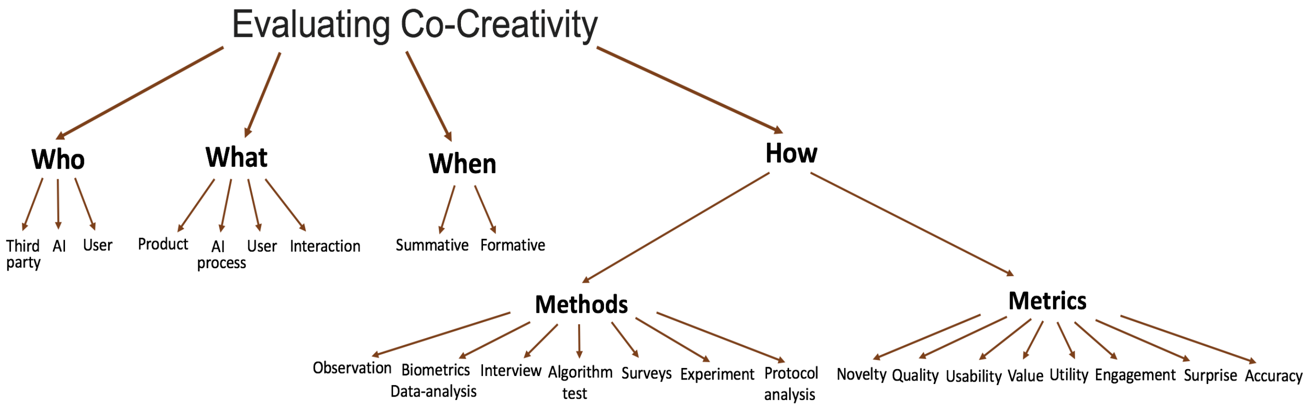


Figure 1: A hierarchical tree of evaluating creativity in computational co-creative systems

which the designer was asked to talk while designing (concurrent) or while viewing a recording of their design task (retrospective). More recently, a design protocol is associated with any sequential recording of a design task, including speech, gesture, body movement, facial expressions, dialogue and digital actions. When evaluating the Drawing Apprentice where users were asked to collaborate on drawing tasks, users were asked to view a recording of themselves designing and describe their thought processes for each action, which were then categorized according to the coding scheme (Davis et al. 2015). This video walkthrough is an example of a retrospective protocol.

Surveys are a method of obtaining data from the users of co-creative system that are much more scalable but less rich than protocol analysis. Surveys can take different forms, but their common goal is to obtain insight into user perceptions of the creative system and the creative tasks. This can include system usability, self-reflection, evaluation of the output, and evaluation of the system's processes. An example of survey data can be found in Sentient Sketchbook (Yannakakis, Liapis, and Alexopoulos 2014), where users were asked about the usability of the game level design tool. In that study users were generally positive about the tool's interface and their interactions with the co-creative system.

Interviews are a qualitative method for evaluating user perceptions of co-creative systems, providing an interpretive alternative to the quantitative and empirical protocol analysis methods. More specifically, these are typically semi-structured interviews, a method common in the social sciences and in human-computer interaction research as a way to elicit rich and nuanced perceptions from small groups of users. In ViewPoints AI (Jacob and Magerko 2015), interviews showed that users expected the agent to respond to each of their movements in real-time and were disappointed on the occasions where it did not.

Observational methods are another common evaluation method. Observing creative tasks without intervening or pre-committing to a specific coding scheme enables investigation of a broad range of behaviors. Examples from co-creative systems include Shimon (Hoffman and Weinberg 2010), where observation showed that the source of inspiration for the current moment of performance alternated between the human and the robot player.

The last user study method that has been successfully employed in the study of creativity support tools, but not yet applied to co-creative systems, is the use of biometric data to quantify human creativity. For example, Carroll and Latulipe (2012) utilized electroencephalogram (EEG) to measure neural signals during a creative task. This work sought to measure 'in-the-moment creativity' (ITMC), which is defined as periods of heightened creativity during the creative process. The EEG data was combined with self-report data about the user's creative state to triangulate when users where experiencing moments of high creativity. This study demonstrates the potential for biometric data to be applied to co-creative systems to help quantify user creativity while interacting with the system.

In addition to user studies, researchers have also tested the algorithms themselves to determine their efficacy. This testing process validates the algorithms and models used by the AI agent employed in the co-creative system. For example, Singh et al. (2017) performs a validation test on the object recognition and generation algorithms used in a co-creative drawing application. This type of validation is common in the machine learning literature to test the effectiveness of the algorithm. In a co-creative context, this information can be used to tweak the algorithm to better suit the needs of the co-creative system.

Metrics The set of metrics for developing computational models for evaluating creativity is very broad, including those defined in (Lamb, Brown, and Clarke 2018; Grace et al. 2015; Ritchie 2007; Wiggins 2006). In response to a focus on novelty and value as the hallmark of creativity that started as early as (Newell, Shaw, and Simon 1959), Maher and Fisher (2012) add a third dimension called surprise, which quantifies how unexpected the creative product is given the sequence of decisions or products that have recently occurred.

Pease and Colton (2011) introduce two different levels for evaluation: cultural value of the outcome (a measure of product) and the complexity of the system's behavior (a measure of process). (França et al. 2016) argue that evaluating computational creativity should be domain independent. They introduces a metric, called Regent-Dependent Creativity (RDC), in which generated artifacts are represented as dependency pairs. RDC measures novelty and value within this structure.

In more recent literature, researchers aim at operationalizing creativity by building computational models. Agres et al. (2015) introduces a computational linguistic model that maps word representations into a conceptual space. The model is based on word co-occurrence in the context of music and poetry. In order to validate the accuracy of the model, user responses to word association is also recorded and compared with the computational model results. Grace et al. (2015) introduces a probabilistic model in order to compute the surprise value in the domain of mobile devices. The model captures the degree of unexpectedness of the observed artifact. These models imply that the less likely an event or combination of events occurs the more likely it is to be surprising.

One important metric that makes co-creative systems different from other computational creativity systems is the engagement of the user with the system. In Viewpoints AI (Jacob and Magerko 2015) the engagement of users was evaluated qualitatively, and was found to be highly positive. In the Sentient Sketchbook (Yannakakis, Liapis, and Alexopoulos 2014) two metrics are developed: perceived usefulness and perceived quality. In 3Buddy (Lucas and Martinho 2017) metrics for utility and efficiency are developed. Many co-creative systems also measure usability, including ViewPoints AI (Jacob and Magerko 2015) and the Sentient Sketchbook (Yannakakis, Liapis, and Alexopoulos 2014).

One final family of metrics applied in co-creative metrics are those derived from accuracy, or more specifically the degree to which generated output matches a reference dataset. These measures often originate from machine learning, where accuracy is a central concern. An example of this is from a recent extension of the Drawing Apprentice system (Singh et al. 2017), in which the classification accuracy and generation loss of their model is reported on two different public datasets.

Case Studies of Co-creative Evaluation

In this section we focus on how the evaluation is performed in different co-creative systems. Table 1 summarizes the above questions for six example systems.

Evaluating creativity in the Drawing Apprentice

In the Drawing Apprentice, several evaluation methods have been deployed, including both formative and summative user studies. Participants are first introduced to the unique features of the Drawing Apprentice system. As an example of formative evaluation, users are asked to rate sketches generated by the agent (like or dislike). This voting occurs at iterative steps when the agent responds to the user's input during the task. For a summative evaluation of co-creativity, a combination of retrospective protocol analysis, interviews, and surveys were performed. Participants were asked to work with drawing apprentice for 12 minutes in two different sessions. In one session, they interact with the actual system and in the other they interact with a "Wizard of Oz" substitute (i.e. a fake system with a hidden human controller). After the task was complete, participants watched a video recording of their interaction and described what they were thinking at each point in the video during a retrospective protocol analysis. Then, participants were asked about their experiences through both interviews and surveys. The results show that the agent is able to coordinate with the user up to a certain degree as well as contributing to the user's drawing.

In more recent work, a machine learning model called an Auxiliary Classifier Variational AutoEncoder (AC-VAE) was added to the co-creative system that allows the agent to classify and generate input images simultaneously in real time (Singh et al. 2017). In this work, the evaluation is done offline through two metrics: classification accuracy and generation loss. Both can be considered measures of value: the degree to which the system is able to categorize sketches made by the user, and the degree to which it is able to produce sketches that are similar to the user's sketch. Results are reported on two different public datasets in order to compare the accuracy of the AC-VAE model to other existing models. Their integration into the co-creative system and their impacts on user behavior and perception of creativity are still under development.

The formative evaluation of Drawing Apprentice leverages the voting system used by the machine learning algorithm in the system. This approach is interesting because it provides a method of evaluating how the agent is performing throughout the session without interrupting the creative flow of the user. It is possible to count how many times users clicked like/dislike, but this method is also unreliable as users do not have to use the voting at all. To get a more holistic understanding of the user's creative experience, the authors employed a retrospective protocol where participants watched their creative process and explained their thoughts. These videos can then be coded to understand themes and trends in the interaction. When supplemented with interviews and surveys, this type of user study can sketch an accurate description of the user's experience with the system. However, this analysis did not include a summative evaluation of the creative output of the interaction, which would help evaluate the relative creativity of both user and system.

Evaluating creativity in ViewPoints AI

The evaluation of this system is done by the users in a public space through a summative and formative user study. Participants are first presented with the prototype of the system and introduced to the features of the system through a demonstration of how to interact with the system. They are then asked to interact with the system without the aid of the researchers. During the interaction, researchers observed how participants interacted with the

System	Who	When	How (Metric)	How (Method)	What
Drawing Apprentice	AI	Summative	Classification Accuracy & Generation Loss	Algorithm testing	Product & Interactive experience
	Users	Formative	Usability	Voting (like, dislike), Survey data & Retrospective protocol analysis	
ViewPoints AI	Users	Formative & Summative	Engagement & Usability	Observation	Product & Interactive experience
Sentient Sketchbook	Experts	Formative & Summative	Usefulness, Quality & Usability	Protocol Analysis, Survey Data, Interview Experiment and Observation	Product
3Buddy	Users & Experts	Summative	Utility & Efficiency	Survey Data, Interview Experiment & Observation	Product
CAHOOTS	Users	Summative	Usability	Survey Data & Experiment	Product
SHIMON	Users	Summative	Engagement	Observation	Product

Table 1: Answers to questions in section three for six different co-creative systems. Note that two studies involving the Drawing Apprentice were published, using different evaluation methods.

system (formative evaluation). After the interaction, participants provided feedback about their experiences (summative evaluation). The results show that users gave positive comments in terms of both the concept and the visual aesthetic of the system. The task observations show that the engagement of the users with the system was highly positive. However, participants were not always able to understand the intentions of the AI agent, with some participants not even understanding that the system was co-creative at all. This highlights the need for AI agents to produce responses that are both similar and different enough to the user's movement. Another finding was that users expect immediate responses during turn-based interaction.

Evaluating creativity in the Sentient Sketchbook

The evaluation of this system is done by the experts through formative and summative user studies. During this study the usability of the sentient sketchbook, game level design tool, is assessed. The evaluation is done online by sending the participants an email and receiving feedback via email as well. The study recruited five users to perform 24 different design sessions. Overall, feedback about the usability of the system were positive.

For the summative evaluation of the creativity of the system, the evaluation is based on two metrics: degree of usefulness of the co-creative tool and quality of their interaction during the process. The first metric, degree of usefulness, refers to usability of design suggestions in different sessions. Based on the user feedback there were cases where the design suggestions were not useful. Particularly most design suggestions were selected in the beginning of the co-creative process. On the other hand, quality of user interaction refers to the impact of the design suggestions on the creative process. In each session, the map instance is shown sequentially based on the user's action. The patterns of the user actions indicate that they prefer a symmetric map both during and after the process.

For the formative evaluation of the creative system, the authors reviewed the user interaction logs from the Sentient Sketchbook system. Each step of the creative process resulted in a slice of what the authors refer to as the 'creation path' that visually depicts the user's journey of creating a game level from start to finish. The authors investigate this formative data to identify different patterns and trends during the user's interaction process.

Evaluating creativity in 3Buddy

The evaluation of this system is done by both the users and the experts through summative user studies. Users are asked to give a value to the two metrics of evaluation called utility and efficiency. Utility refers to the ability of the system to contribute useful content. Efficiency refers to the degree in which the co-creative tool can produce useful and coherent content.

The user study conducted to evaluate 3Buddy (both surveys and interview questions) focused on how easy the system was to use, including utilizing the various features of the tool. This type of usability analysis is interesting to evaluate the effectiveness of the tool, but it does not reveal insights about the creativity of the user or the system throughout the co-creation process. To further augment this type of investigation, the authors could employ a protocol analysis to observe the user and system behavior through time, similar to the concept of 'creation path' introduced by (Yannakakis, Liapis, and Alexopoulos 2014).

Evaluating creativity in CAHOOTS

The evaluation of this system is done by the users through summative controlled user studies. In order to test the usability of the system, participants are first introduced to the design of the system and are asked to perform a conversation for 10 minutes. Then the pairs of participants are presented with three variants of the system and are asked to chat for 10 minutes. By the end of the study, participants are required to fill out a survey in order to evaluate both the conversation and the system. The results show that participants were able to be involved in the conversation as well as finding the conversation to be funny. They also felt close to their partner as well as being able to express their sense of humor during the conversation. In order to address the qualitative analysis, the participants feedback on both prototyping and experimental phases is gathered. The results show that the feedback was positive.

The experiments conducted to evaluate CAHOOTS focused on usability and enjoyment, comparing it to standard text messaging applications. This type of usability analysis can reveal user satisfaction with the system, but the authors did not discuss how to evaluate the creativity of the system. Additional considerations could investigate how the suggestions of the system influence the creativity of the user and how creative the user thinks the system is in different conversational contexts.

Evaluating creativity in SHIMON

The evaluation of this system is done through a live performance with 160 attendants for seven minutes through a summative user studies. In this performance the robot, Shimon, with the gesture-based improvisation is shown to the audience. During the performance, a human pianist performs an opening phrase, then the robot detects the phrase and responds with preliminary gestures. This performance has three segments: The first is an open-ended collaboration between the human pianist and the robot player, Shimon. In the second phase, the robot plays in opportunistic overlay improvisation. In the last phase the robot uses a rhythmic phrase-matching improvisation.

The authors describe a performance-based evaluation of the SHIMON system during which an audience observed the system in action as it was improvising with users. The evaluation included analyzing how the system behaved during the performance as well as audience reactions to the performance. The results of the authors analysis show that there was an alternating inspiration between the human and the robotic player. The authors also note that a video recording of the performance was widely acclaimed by the press and viewed over 40,000 times. This type of evaluation falls under the 'observation' category in our framework because the authors were working to understand how the audience perceived the performance. In the future the authors are interested to evaluate the system's gestures as well as the effect of the robotic player on band-members and audience.

Conclusions

This paper provides a framework for evaluating creativity in computational co-creative systems. The framework provides a structure for comparing the evaluation of co-creative systems across specific examples and implementations, as well comparing to other types of creative systems, such as autonomous creative systems and creativity support tools. By asking questions such as who evaluates, when does evaluation occur, what is evaluated, and how evaluation is performed, we can broaden the scope of evaluation studies and apply methods from one area of computational creativity to another area.

In our study of evaluation in existing co-creative systems we found a dominant focus on evaluating the user experience and the product of the experience. This demonstrates that many existing co-creative systems extend creativity support tools to include more pro-active contributions from the computational system.

Unlike creativity support tools, co-creative systems have the potential for self-evaluation by embedding a self-awareness of the creativity of the AI agent. With a focus on evaluating the creativity of the AI agent, the computational contributions to the collaboration can be directed by its perception of the creative product. The capacity for self-evaluation can guide users towards or away from particular regions of the space of possibilities intentionally based on the the AI agent's concept of creativity.

Unlike autonomous creative systems, co-creative systems have the benefit of human interaction that can introduce the human perception and evaluation of the creative product during the process. Such a co-creative system requires flexibility, interruptibility, and transparency. Different strategies for achieving co-creativity include turn taking, framing, and explainable AI techniques. These strategies highlight the importance of accommodating when the AI agent has a particular intent or goal that is at odds with the user. Co-creative systems containing agents as partners will require communication of rationale and justification in order to achieve the kind of co-creativity sessions we would expect when it is among people only.

Unlike fully autonomous creative systems and creativity support tools, the creative process used by co-creative systems is not the result of a single agent, instead it is a collaboration. This means existing approaches to evaluating computational creativity or HCI approaches to evaluate creativity support are insufficient. This identifies a new focus for research in computational creativity to study how creativity can be evaluated in human/AI collaboration with the combination and intersection of usability and creativity metrics. Evaluative methods and metrics are a step towards self-aware and intentional co-creative agents.

References

Agres, K.; McGregor, S.; Purver, M.; and Wiggins, G. A. 2015. Conceptualizing Creativity: From Distributional Semantics to Conceptual Spaces. In *ICCC*, 118–125.

Besemer, S. P., and O'Quin, K. 1999. Confirming the three-factor creative product analysis matrix model in an American sample. *Creativity Research Journal* 12(4):287–296.

Carroll, E. A., and Latulipe, C. 2012. Triangulating the personal creative experience: self-report, external judgments, and physiology. In *Proceedings of Graphics Interface 2012*, 53–60. Canadian Information Processing Society.

Colton, S.; Halskov, J.; Ventura, D.; Gouldstone, I.; Cook, M.; and Ferrer, B. P. 2015. The Painting Fool Sees! New Projects with the Automated Painter. In *ICCC*, 189–196.

Colton, S. 2008. Creativity Versus the Perception of Creativity in Computational Systems. In *AAAI spring symposium: creative intelligent systems*, volume 8.

Compton, K., and Mateas, M. 2015. Casual creators. In *ICCC*, 228–235.

Cox, M. T., and Raja, A. 2011. *Metareasoning: Thinking about thinking*. MIT Press.

Cropley, D. H.; Cropley, A.; et al. 2005. Engineering creativity: A systems concept of functional creativity. *Creativity across domains: Faces of the muse* 169–185.

Davis, N.; Hsiao, C.-P.; Popova, Y.; and Magerko, B. 2015. An enactive model of creativity for computational collaboration and co-creation. In *Creativity in the Digital Age*. Springer. 109–133.

Davis, N.; Hsiao, C.-P.; Singh, K. Y.; Lin, B.; and Magerko, B. 2017. Creative sense-making: Quantifying interaction dynamics in co-creation. In *Proceedings of the 2017 ACM SIGCHI Conference on Creativity and Cognition*, 356–366. ACM.

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*.

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.

Grace, K., and Maher, M. L. 2016. Surprise-Triggered Reformulation of Design Goals. In *AAAI*, 3726–3732.

Grace, K.; Maher, M. L.; Fisher, D.; and Brady, K. 2015. Data-intensive evaluation of design creativity using novelty, value, and surprise. *International Journal of Design Creativity and Innovation* 3(3-4):125–147.

Hoffman, G., and Weinberg, G. 2010. Gesture-based human-robot jazz improvisation. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, 582–587. IEEE.

Jacob, M., and Magerko, B. 2015. Viewpoints AI. In *Proceedings of the 2015 ACM SIGCHI Conference on Creativity and Cognition*, 361–362. ACM.

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.

Karimi, P.; Davis, N.; Grace, K.; and Maher, M. L. 2018. Deep Learning for Identifying Potential Conceptual Shifts for Co-creative Drawing. *arXiv preprint arXiv:1801.00723*.

Lamb, C.; Brown, D. G.; and Clarke, C. L. 2018. Evaluating Computational Creativity: An Interdisciplinary Tutorial. *ACM Computing Surveys (CSUR)* 51(2):28.

Lucas, P., and Martinho, C. 2017. Stay Awhile and Listen to 3Buddy, a Co-creative Level Design Support Tool. In *eighth International Conference on Computational Creativity, ICCC, Atlanta*.

Maher, M. L., and Fisher, D. H. 2012. Using AI to evaluate creative designs. In *DS 73-1 Proceedings of the 2nd International Conference on Design Creativity Volume 1*.

Newell, A.; Shaw, J. C.; and Simon, H. A. 1959. *The processes of creative thinking*. Rand Corporation Santa Monica, CA.

Pease, A., and Colton, S. 2011. On impact and evaluation in computational creativity: A discussion of the Turing test and an alternative proposal. In *Proceedings of the AISB symposium on AI and Philosophy*, 39.

Ritchie, G. 2007. Some empirical criteria for attributing creativity to a computer program. *Minds and Machines* 17(1):67–99.

Saunders, R., and Gero, J. S. 2001. Artificial creativity: A synthetic approach to the study of creative behaviour. *Computational and Cognitive Models of Creative Design V, Key Centre of Design Computing and Cognition, University of Sydney, Sydney* 113–139.

Schmidhuber, J. 2008. Driven by compression progress: A simple principle explains essential aspects of subjective beauty, novelty, surprise, interestingness, attention, curiosity, creativity, art, science, music, jokes. In *Workshop on Anticipatory Behavior in Adaptive Learning Systems*, 48–76. Springer.

Shneiderman, B. 2007. Creativity support tools: Accelerating discovery and innovation. *Communications of the ACM* 50(12):20–32.

Singh, K. Y.; Davis, N.; Hsiao, C.-P.; Macias, R.; Lin, B.; and Magerko, B. 2017. Unified Classification and Generation Networks for Co-Creative Systems. In *eighth International Conference on Computational Creativity, ICCC, Atlanta*.

Wen, M.; Baym, N.; Tamuz, O.; Teevan, J.; Dumais, S. T.; and Kalai, A. 2015. OMG UR Funny! Computer-Aided Humor with an Application to Chat. In *ICCC*, 86–93.

Wiggins, G. A. 2006. A preliminary framework for description, analysis and comparison of creative systems. *Knowledge-Based Systems* 19(7):449–458.

Yannakakis, G. N.; Liapis, A.; and Alexopoulos, C. 2014. Mixed-initiative co-creativity. In *FDG*.

Is Computational Creativity Domain General?

Róisín Loughran and **Michael O'Neill**
Natural Computing Research and Applications Group (NCRA)
University College Dublin
Ireland
roisin.loughran@ucd.ie

Abstract

The question as to whether creativity is domain general or domain specific is one of the most enduring and controversial topics in the field. Yet the importance or relevance of the chosen application domain has not been considered in the related field of computational creativity (CC). A recent study at ICCC demonstrated that the range of applications considered in the study of CC has been diverse with more novel topics being considered as the field progresses (Loughran and O'Neill 2017a). As the field grows, we propose that we need to consider the relevance of the application domain and any potential role or effect the choice of the domain may have on the outcome of the designed system. In this paper, we review what it means for CC to be domain-general. We consider the domain-dependence of creativity in human studies and what implications, if any, may arise from the choice of application domain in CC studies. We conclude that this is a multi-faceted question and that a simple yes or no answer may not be possible to acquire or sensible to suggest.

Introduction

Computational systems that attempt to simulate, portray or genuinely exhibit creativity typically do so in a given application domain. A generative creative system can create some novel artefact such as a melody, piece of artwork or joke to be evaluated in order to ascertain the level of creativity exhibited by the system. Despite the diversity in possible domains there have been trends in the topics considered at ICCC with systems based around music and Natural Language Processing (NLP) remaining popular over the years (Loughran and O'Neill 2017a). Even so, the same study showed that there is a steady increase in systems based in novel application domains not considered before at ICCC. In another study that year on 'How to Build a CC System' it is proposed that the first step in the process is: choose a domain *D* (Ventura 2017). If this model is followed then all subsequent steps in the building of the system are dependent on this first step. If the application domain is such a fundamental choice and yet is so diverse — and getting increasingly so — between studies, it raises the question as to whether or not the chosen application domain has an effect on the potential creativity that could be displayed by the system.

The question, as posed in our title, requires clarification. The term Computational Creativity (CC) is defined as (Colton, Wiggins, and others 2012):

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

Thus CC refers to a field of study. Asking whether or not a field is domain-general is a much broader and more complex question than asking if a computational system is domain-general. In this paper we will mostly consider the latter question: Is the creativity demonstrated by or aspired to in a given computational system dependent on the application domain within which the system is placed? Even this more specific question raises many issues to consider: Is creativity itself domain-general? Is the creativity of a system dependent on the choice of domain by the programmer? If these questions were found to be true it could raise the issue as to whether or not the creative capacity of the system is dependent on the creativity of the programmer. Furthermore, we may need to consider if artefacts produced in certain domains *appear* more creative than others and if so does this make such domains more suitable for study than others? Do impressive results in traditionally 'creative' domains such as music or art give the impression of higher creative levels than those in more simple domains? A focus on impressive results without considering the underlying cognition of the creativity that produced said results can lead to a misunderstanding between weak CC — that which merely simulates human creativity. and strong CC — systems that exhibit autonomous creativity (al Rifaie and Bishop 2015). There is a place in the field for systems that focus on both weak and strong CC, but it is vitally important to be clear as to which is under consideration.

This paper examines the question of domain generality of systems developed within the field of CC. We review the domain-generality of creativity in human studies and consider what differences may lie in studying creativity from a computational standpoint. We review the types of creativity proposed in the literature and consider how they are approached by CC studies in various application domains.

Human Creativity Studies

The current accepted definition of CC, given above, is in terms of creativity itself. The reason for this circularity lies in the fact that creativity remans to be such a difficult concept to define. It has been argued that whether creativity is domain dependent or domain general is one of the most controversial issues in (human) creativity research (Plucker 2004). For this reason, we consider general creativity — the manner in which it has been defined and evaluated and the question as to whether it is domain specific or domain general, before considering its relationship to computation.

Definition of Creativity

The standard definition of creativity is succinct: 'Creativity requires both originality and effectiveness' (Runco and Jaeger 2012). But unfortunately, this is far from the only definition that is used. One of the main problems in the scientific study of creativity is that there have been so many different definitions proposed over the years. It has been stated that there exist over a hundred definitions for creativity within the relative literature (Meusburger 2009). However, considering the subject of creativity has been studied in many subject fields including philosophy, psychology, education, sociology plus all application and technical fields, it is likely that the number is significantly higher than this. Even different dictionary definitions of the word contain discrepancies. The Oxford English dictionary currently defines creativity as 'The use of imagination or original ideas to create something; inventiveness'[1], whereas the Cambridge English dictionary define it as 'The ability to produce original and unusual ideas, or to make something new or imaginative'[2]. In a study on the history of creativity for AI research it has been stated that 'Creativity needs creativity to explain itself'(Still and d'Inverno 2016). Hence even in general creativity, as in the definition of CC, the term is self-referential.

Despite the number of definitions that have been proposed, the common elements that have been present in any accepted definition are based in novelty (or originality) and value (or effectiveness). While the roots of the study of creativity began to emerge in the 1930s-1950s (Runco and Jaeger 2012) the first definition to include these elements was given by Stein: 'The creative work is a novel work that is accepted as tenable or useful or satisfying by a group in some point in time' (Stein 1953). More recent works that attempt to define or evaluate creativity do so by focussing on the two aspects of novelty and value (Ritchie 2001; Boden 2004). Value can be attributed to a concrete artefact or to a more abstract concept theory or interpretation. Novelty can refer to ideas that are new to the individual, know as Psychological (P) Creative, or those that are novel to the world — Historical (H) Creative. By this reasoning H-Creativity is a special case of P-Creativity (Boden 2004).

Csikszentmihalyi similarly separated the idea of creativity to the individual (P-creative) and to a culture (H-creative) in proposing the idea of 'Big C' Creativity (Csikszentmihalyi 2013). He posits that such a version of creativity cannot be experienced purely by an individual, but must have an influence on some aspect of culture. He considers that this Creativity can only be found in the interrelations between three parts of a system:

- Domain: a set of symbolic rules and procedures;
- Field: those who decide what is novel within the domain;
- Person: those who undertake the creative act or idea.

Only within the interplay of these parts can Creativity be exhibited. Thus he sees Creativity as an act that changes an existing domain or transforms a given domain into another domain. Boden has also proposed three distinct types of creativity: combinational, explorational and transformational (Boden 1998). She proposed that transformational creativity, which transforms the space within which one is searching, offers the most opportunity for discovery. Hence, transformational H-creativity is akin to Csikszentmihalyi's Creativity ('Big-C'), but due to the transformation of domains and requirement of historical domain knowledge, such creativity will be difficult to evaluate.

Evaluating Creativity

The evaluation or measurement of creativity is often reported in relation to assessing creative ability, often in children, through a variety of 'paper-and-pencil' tests (Cropley 2000). Although a large variety of such tests exist, Cropley proposed concentrating on those developed during the 'modern creativity era' as first introduced by Guilford (Guilford 1950). He reviewed and organised such tests to reveal four dimensions relating to elements of creativity: product, process, motivation and personality/ability. From an analysis of a large number of tests he observed that creativity tests did not have as high a predictive value of success as traditional IQ tests. Cropley also stated a preference for models that encompass both thinking and personality such as the Test of Creative Thinking: Divergent Production (TCP-DP) (Urban and Jellen 2005) over those centred purely on divergent thinking (DT). DT tests, however have been used extensively as an indicator for creative potential (Runco and Acar 2012). While DT is acknowledged as different to creative thinking, psychometric tests have suggested that these tests can provide predictors for the potential of creative thinking. While Guilford tied divergent production to creative potential, Runco explicitly states that DT is not synonymous with creativity, but that tests based on this theory can indicate a potential for creative problem solving.

Although there have been many tests for identifying creative thinking, no one test has been agreed on as the most general or best. Furthermore, these DT style tests are not discussed in terms of application domain but rather in terms of general creative thinking potential. Nevertheless, the importance of the given application domain on creativity and creative ability has been discussed at length in the field.

Domain Specificity of Creativity

Whether creativity is domain specific or domain general has been a hotly debated topic in creativity research for a number of years. In general creativity, this amounts

[1] https://en.oxforddictionaries.com/definition/creativity

[2] https://dictionary.cambridge.org/dictionary/english/creativity

to asking whether a person, process or product is considered creative only within one domain or across multiple domains and furthermore whether training in one domain can increase ability in another creative domain. There are strong opinions on both sides with some arguing for domain generality e.g. (Runco 1987) with others maintaining the domain specificity of creativity e.g. (Baer 1998; 2010). While the discussion continues, the dominant perspective appears to be leaning towards domain specificity. Even if a consensus cannot be reached, Baer still argues that it is better to assume specificity over generality (Baer 1998). He proposes that in assuming specificity, nothing is lost by training a subject in their specific domain even if it turns out that that creativity is domain general. Conversly if domain generality is assumed but domain specificity is the reality, much effort could be wasted in teaching and learning based on general domain creative-thinking tasks. Plucker and Beghetto further argue that too much focus on either position may hinder creativity. They conceptualise a model indicating a focus on generality can lead to superficiality whereby one may never fully engage in the creative task, while being too specific could lead to a fixedness whereby the goal is never satisfied (Plucker and Beghetto 2004). They conclude that creativity is most likely domain general, but that it can appear domain specific and that ultimately it is not beneficial to dwell on the concept

Whether or not creativity is domain-specific as defined above, it is evident that creative domains exist and are important to us. As Csikszentmihalyi has stated: 'The existence of domains is probably the best evidence of human creativity' (Csikszentmihalyi 2013). In contrast to many other living organisms, we as humans can choose from a number of responses when presented with a given stimulus; a flower will turn towards the light of the sun, but when we encounter such a flower we may choose to focus on visual art and paint it, we may be inspired to write a poem about it, or we may pass it by with barely a glance while focussing on other matters. An initial choice can lead to a decision to work in more specified sub-domains; if we want to make a visual artistic rendition of the flower do we paint, use charcoal or printing? Our individual responses to the world may be different, there is no standard domain in which me must work. We choose which specialist domain to focus on and the more we work on a domain, the more specialised we become. Animals other than humans are capable of actions other than the predefined responses of the above flower, however. Much creative and playful interactions have been observed in the social and exploratory behaviours or animals (O'Hara and Auersperg 2017). While these may not result in behaviour considered as traditionally creative as that of humans, creativity in nature has been noted to be heavily influenced by similar such social interactions (Saunders and Bown 2015).

The Computational Comparison

A person must first find their passion to consider working in any given domain. Only through years of study can they then choose their own sub-domain and find their footing to build on what has been done in order to make any creative contribution. Can an autonomous creative system make such decisions before it undertakes a work? The decision as to what domain to work in is invariably decided on *a priori* by the programmer: we plan to write music generation programs or art generation programs. Is it possible to write a creative program without first specifying the domain —- and by specifying the domain have we paradoxically removed the potential for the system to intentionally display creativity?

While we may accept that the lack of clarity of a definition for creativity leads to the circularity in the CC definition, it does mean that this CC definition is reliant on an ill-defined concept. Computational measures, by definition, are based on the idea of enumeration — an exact process. Thus what we are trying to achieve in CC is to enumerate that which we cannot define. To some, this may be the best argument against applying computation to creativity; if we cannot define the concept and therefore cannot measure the concept, how can we expect computers to enumerate, generate or imitate the concept? Such an argument is overly defeatist however. Creativity is not a divine ability afforded to only a lucky talented few, it is merely a feature of human intelligence (Boden 1998). Much of the original theory on creativity was based on determining if it was distinct from intelligence (Runco 2014). AI systems are becoming increasingly important in our modern day society. With the push towards a general AI, it is imperative we recognise creativity as an important aspect of intelligence and do not merely dismiss the idea of computing it as too much of a challenge.

We have established that creativity requires novelty and value, but there is one extra important aspect that must be considered, particularly when considering autonomous creative systems: the aspect of intent. The requirement of intent is rarely addressed in humn creativity; presumably a person who is creating an artefact is doing so intentionally. Many recent CC studies however have stated that for creativity to be present, an agent must exhibit novelty, value and intentionality e.g. (Ventura 2017). Novelty and value are important but once we consider creation by a computative system, this idea of intent or ownership becomes equally important. If a system generates a joke — does it need to have intended to do so in order to display creativity? What would such an intent even mean — was the system attempting to make us laugh? This level of invoking an emotional response from a computative system is not possible yet, but for creativity to be displayed the system should provide evidence on some level of intending to produce its output. This issue of intent has raised discussion in recent years. (Guckelsberger, Salge, and Colton 2017) considered a non-anthropocentric model by adopting an enactive AI framework finding that CC systems that focussed on human creativity typically cannot provide a reason for intent as they lack intrinsic goal-ownership.

CC Evaluation A lack of evaluation has been noted many times throughout the development of the field of CC (Boden 1998; Cardoso, Veale, and Wiggins 2009; Jordanous 2011). This lack of evaluation could result in undermining any scientific progress of the field, resulting in a stricter focus towards evaluation within papers and the development of a number of frameworks according to which CC systems should be evaluated. Over the past decade a number of evaluation frameworks have been proposed including a set of empirical criteria (Ritchie 2007), the Creative Tripod (Colton 2008), numerous Turing-style tests (Ariza 2009) and the Standardised Procedure for Evaluating Creative Systems (SPECS) (Jordanous 2012). Ventura gave a series of milestones that a system must surpass in order to be in the 'realm' of creativity (Ventura 2016). He posits that systems should exhibit more than just randomisation, plagiarisation, memorisation, generalisation, filtration and inception to avoid the 'mere generation' trap, but concludes that the location of actual creativity is still somewhat out of reach from the computative community.

One test for creativity that is focussed on autonomy is the Lovelace Test (LT) (Bringsjord, Bello, and Ferrucci 2003). This involves an artificial agent A, its output o and its human architect H. Simply put, the test is passed if H cannot explain how A produced o. It is important to note that this test is not about predicting results but explaining how they came about. Any programmer who can explain their written code can in theory explain how the their agent produced its output. The only situation in which this LT can be passed is one whereby the programmer cannot explain what they have written. In reality, if a system consists of multiple modules that interact to produce a final output, one could argue that not one individual programmer could explain the whole system, but theoretically their combined knowledge should be able to, and as such this scenario is merely an increase in complexity rather than a true solution to the test. As it stands, the LT does not appear to be passable, regardless of what application domain it is applied to.

By necessity, evaluation of the output or artefacts produced by any creative system is domain-dependent; if evaluation is performed on an artefact, then the domain of said artefact is imperative to the judgment. The field of CC is based on systems rather than artefacts however. While there still remains a tendency to evaluate purely on the final produced artefacts, such assumptions can lead to limitations within evaluations and hence in the growth of the field in general (Loughran and O'Neill 2017b). It would be beneficial instead to be able to make an evaluation of the running system, where this is more appropriate. Such a judgement should not be dependent on the application domain, but could in theory be dependent on the type of algorithm used in the development of the system. The LT is an example of such a metric as it is based on how the artefact is produced — not merely what is produced. Similarly the SPECS method could incorporate such a method as the definition of creativity and the standards used to evaluate it are both specified as part of the method. Hence, the question as to whether or not the evaluation of creativity is dependent on application domain is more dependent on the definition of the evaluation rather than the definition of creativity.

Application Domains in a CC System

The design and implementation of any computational system involves a number of steps. Such a design may start with a choice of algorithm, analysis of data, a focus on inner workings or any high level consideration of the system. A recent proposal for the first step in creating a CC system is in choosing the application domain (Ventura 2017). Using this framework, once that is chosen, all representations both at a genotypic (internal) and phenotypic (external) level can be determined. Therefore, in the development or planning of a system, the application cannot be disregarded; what the system creates is quite often the initial purpose of making the system for many people. People naturally tend to work with systems that operate within a field that they themselves have domain knowledge, and they often stay within one application domain; musicians tend to make musical systems and artists work with visual systems. This results from personal interest but should not have any direct bearing on the potential creativity within the system if, that is, we can consider creativity to be domain general. This does ensure, however, that the initial intent in creating the system lies solely with the programmer rather than the system. Even if we adopt this framework and assume that it is acceptable to choose the application domain as the first step of building a system, the system should subsequently display some level of intentionality in creating its output.

Concept

The analysis of application domains considered throughout ICCC listed 'Concept' as one singular domain (Loughran and O'Neill 2017a). In essence, however, many CC systems could be reduced to Concept, regardless of the given application domain. A story-telling system, such as Mexica (y Pérez 2015), may make use of NLP but this involves more than mere syntactic analysis of words. Yes, the text must make grammatical sense, but the meaning behind the words and the arch that the story follows — interaction between characters, emergence of themes, building and subsequent release of tensions — are what will engage the reader. These higher level features are less about the domain (NLP) as much as they are about the concept behind them. Similarly, in systems that deal with other domains such as music, games etc. the creativity within the system could be contained in the underlying concept, but this concept is wrapped up in layers of increasingly complicated representation. The application domain is the public front to such systems, and it can catch someone's eye, but is not necessarily where the creativity lies. While systems that generate very impressive aesthetic outputs may rely heavily on domain knowledge, others can be reduced to an underlying concept for more insightful understanding. The importance of the underlying concept within any CC system can be related to the given application domain.

Another particular topic of of interest often considered in CC studies is that of analogy. Studies in analogy generally use the written word and so could be broadly put into the domain of NLP. But again it is not the semantic understanding of the words that is under consideration, rather the interplay of the underlying meaning behind those words between two specific concepts. As such all studies in analogy consider two conceptual domains and the transition between them. Similarly, studies that consider conceptual blending (Fauconnier and Turner 2003) draw from more than one original domain. Conceptual blending integrates two or more mental spaces in order to create something in a new blended space.

CC studies that consider concept, analogy or blending do not always have distinct boundaries in regards to application domain. In such cases, the domain may not be simple to define and hence can appear to be more general than those studies focussed on specific aesthetic artefacts.

CC Systems in Multiple Domains

Throughout the development of CC there have been a number of systems proposed and studies described that deal with more that one application domain. Some studies may use a specific domain to illustrate a point that is in fact domain-independent such as the theoretical model of creative inspiration proposed in (Wiggins 2012). A number of studies propose the examination of a new general principle and subsequently illustrate the point using a variety of examples from different domains such as considering intrinsic measures of fitness (Cook and Colton 2015), antagonistic and supportive behaviours (Guckelsberger et al. 2016) or multiple facets in preference functions (Bhattacharjya 2016). These studies typically propose a new method of considering or measuring CC, either formally or empirically, and then illustrate these concepts in concrete examples. Such studies may consider multiple domains, but are still limited to those under consideration, rather than generalising across all domains.

Adaptive Systems

The above discussion focusses on generative systems. If a system is created with the purpose of generating 'something' then boundaries must be implemented within which this something will be generated. The limits of these boundaries constitute the domain and thus this domain must be specified by the programmer at the beginning of development. If instead, a system is developed whereby the ability or creativity of the system lies in the modification of an existing artefact or behaviour this may be considered *adaptive creativity* (Bown 2012). Adaptive creativity can be exhibited by flexible systems that adapt not just to their internal composition but respond to external perturbations within dynamic environments. As the domain or function within the system is already established, the explanation, evaluation or fitness of the adaptation should be possible; an adaptive system should be able to evaluate any changes to stay within viability boundaries (Guckelsberger, Salge, and Colton 2017).

If a system displays adaptive creativity, this may be exhibited within a certain domain but such creativity may not be *dependent* on this domain. The creativity emerges through the traversal of the behaviour or artefact being adapted within specified boundaries or limitations. In such a case it would be very difficult to argue that the creativity exhibited is not domain-general; if this creativity is exhibited within one given domain without being developed because of the domain, arguably this could be transferred to another domain to exhibit similar results.

Big and Little Domains

We have established that creativity is not limited to extremely impressive or artistic feats. Creativity is a general aspect of intelligence, as described in Boden's 'P' and Csikszentmihalyi's 'small c' creativity. Yet, often by focussing on systems in specific domains that are associated with talent or passion, such as music, the purpose of the system can subconsciously focus on big-C Creative results. When someone hears that a computer system has written music, the expectations of the quality of the produced is automatically high — why would a system that creates mediocre music be of interest? It becomes difficult to focus on or even appreciate small-c creative achievements when one is effectively working within a big-C domain. In such a big-C domain, much *a priori* domain knowledge is required for a system to generate anything of value, hence the domain becomes important and it can be difficult to disentangle the creativity from the application domain.

If instead, systems were developed in less traditionally creative, aesthetic or artistic domains such as logic or problem solving, it is possible that small-c creativity would be more accessible to identify or study. Besold posited that CC systems belong to one of two families of 'artistic creativity' or 'problem-solving creativity' noting that the latter has stronger links to transformational creativity and is closer to strong creativity than more aesthetic based studies (Besold 2016). He acknowledged the lack of research focussed on computational cognitive systems with general creative capabilities that are mostly independent of a concrete domain, as similarly noted in (Loughran and O'Neill 2017a). If we wish to cut through the domain dependence of CC systems to consider a more general understanding, it is certainly worth considering a stronger focus on the cognitive aspects of computational agents as they undertake problem-solving tasks.

'Humanity' in CC

The current CC definition in (Colton, Wiggins, and others 2012), makes no reference to human opinion. Earlier definitions and discussions on the topic have made reference to human ability however. In Marsden's discussion on Intelligence, Music and Artificiality he discusses the 'intention to perform in a human-like fashion' as one of the two major topics of the paper (Marsden 2013). Ritchie justifies alluding to human-creativity when considering more general (non-human or machine) creativity for two reasons: that this is the established usage and secondly, that doing otherwise would risk circularity in claims about the process (Ritchie 2006). The definition of computational creativity offered in 2006 by Wiggins referred to behaviour of systems which would be 'deemed creative if exhibited by humans' (Wiggins 2006). As late as 2012 Jordanous' definition [3] referred to behaviour 'if observed in humans'.

Despite the lack of the term 'human' in the definition, there remains a lingering tendency to consider human opinion when evaluating or discussing CC systems. (Loughran and O'Neill 2016) have argued against a consistent human-comparison when it comes to evaluating generative musical systems. They proposed that evaluations which focus purely on human-based measurements would automatically be subjected to bias and therefore result in limitations in the development of the field, particularly in the area of au-

[3]quoted from the `computationalcreativity.net` website at that time

tonomous creativity. This is in mind with Guckelsberger's non-anthropocentric method of considering intent in CC and Amabile's earlier warnings of considering attributions of creativity purely as those of the individual (Amabile 1995). If the explicit mention of human opinion is not part of what defines CC, we should be mindful not to let a bias based on human opinion to creep back into evaluations and discussions on what it is to be creative.

Machine vs. Human Capabilities

In considering CC, or any form of AI, we effectively attempt to model some aspects of brain function through computation. Many techniques developed in the field of ML are based on brain function such as connectionism and artificial neural nets. With the dramatic increase in computing power, the capabilities of such systems are likewise increasing. Thus there are two interrelating questions: are we modelling the brain correctly and if so, do we have enough computing power to emulate general human intelligence? While the theory of neural networks has been available for over half a century, it is only in the recent advancement of deep neural nets consisting of millions of neurons, that the computing power of such systems has become clear. These networks still offer a black-box approach to problem-solving however, rendering successfully trained networks difficult to analyse. The training of such networks is furthermore specific to a given task, there does not yet exist an AI of general intelligence. Nevertheless, with the advancement of such technologies it is becoming more feasible that an artificial mind as powerful as our own in on the horizon.

General AI or indeed strong (domain-general) creativity may not be possible until an artificial mind as powerful as that of a human is a reality. Even if the computational power is reached, it is not an absolute that it will model the world in the same manner as we experience it. For the moment, we must be content to work within the current bounded rationality of computer capabilities; we develop the models we can within the current limits. In developing weak creative systems or domain specific tasks we can consider creativity and intelligence from different aspects, thus moving towards a multi-faceted model or understanding of strong general creativity.

Discussion

From an academic standpoint, should we consider creativity by computers in the same manner in which we consider creativity by humans? In defining what creativity actually means it appears that we must, yet in discussing creativity across domains it appears we cannot. In examining what constitutes creativity, the discussion revolves around notions of novelty and value regardless as to who or what displayed the creativity in question; if the term creativity is to refer to a clearly defined concept, then the entity involved should not matter. Discussing domain generality as it is proposed in human studies does not necessarily translate to the computational realm. Many of the human studies on domain generality discuss increasing the ability of a child in one domain from practice or learning in another domain. A similar transfer of knowledge between computational system from one domain to another could only be possible in the presence of general intelligence. While transformational creativity is a well known concept in CC, and knowledge transfer may be possible in certain circumstances, the transfer of knowledge from one domain to another requires a general level of adaptation or intelligence that is not presently available in an AI.

If a system is created for the purpose of generating an aesthetic artefact, as we have established that many — but not all — are, then the intentionality of working within the given application domain is wholly on the programmer rather than within the system. Furthermore, regardless as to how prestigious or accomplished a human creator is at their craft, they suffer from self-doubt and sometimes crippling self-criticism which can ultimately lead to a difficulty in ever finishing a piece of work (Nebel 1988):

> 'A work, finished or not, produces in many artists an aftershock ('choc en routour') of one type or another. As a result, artists are led to reflect upon the work, to change, modify, and even on occasion to destroy it.'

Such self-criticism is considered to be a built-in feature of the artistic personality, and it can lead to feeling of anguish at the thoughts of finishing a piece of work; to a perfectionist a piece may never be truly finished, rather they merely feel they must stop as they can do no more to it. Constant corrections, and an unattainable goal could be programmed into a computational system, but they cannot feel this self-doubt or anguish at their own self-imposed notion of mediocrity. Is such self-criticism necessary for true creativity or in contrast, are such feelings emphasised by authors describing the plight of the tortured artist?

The artists of the Renaissance period may have worked in many domains, but as the years progressed the art world has become more fragmented, more specialised with more subdomains not just in medium but in style and purpose within to place oneself. As expertise within a given application domain has become more specialised, it has been speculated that the emergence of 'Renaissance era' artist — one that exhibited such creativity in many areas as Da Vinci once did — are becoming less likely (Plucker and Beghetto 2004; Csikszentmihalyi 2013). If we no longer expect even the most talented contemporary artists to display such general creativity across multiple domains, is it fair to expect such achievements from computational systems that are under development? Again such arguments are based on the idea of working in Big-C domains, it is important to remember that such opinions propose much less of an issue in CC systems that focus on small-c logical problems.

At times the discussion around creativity has been framed as either creative ability or creative activity. It is important to consider which of these the term 'creativity' is actually referring to. Creative activity is surely domain-dependent, the activity in question must inherently take place within a given domain. But creative ability may be more ambiguous. The definition of creativity given in (Plucker and Beghetto 2004) stresses the interplay between ability and process:

> 'Creativity is the interplay between ability and pro-

cess by which an individual or group produces an outcome or product that is both novel and useful as defined within some social context.'

While they acknowledge that both aspects have been discussed within numerous definitions, they consider that it is in this interplay that creativity lies; regardless of ability, creativity must be enacted through a process.

It has been noted throughout this paper that a critical difference that arises between the discussion of creativity by machines and by humans is that of intentionality. CC is defined in terms of being creative, therefore if the term *creativity* is to be used in the same manner regardless of who or what is exhibiting it, we propose the concept of intentionality should be explicitly stated in the CC definition:

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

Conclusion

We started this article by posing the question 'Is CC domain general?'. Asking whether or not human creativity is domain-general cannot necessarily be treated in the same manner as asking whether or not computational creativity is domain-general. When discussing the former, studies tend to be based on teaching people in one domain and ascertaining the effects such teachings would have on their abilities in other creative domains. This process does not make sense in computational systems unless we consider these systems to have enough general intelligence to understand both domains. Such a system would need both general intelligence and strong creativity. What we should consider is whether or not creativity itself is more suited, emergent or likely in one domain over another — as conducted in computational experiments. A comparison in this manner is not always logistically possible, however. Systems that operate in differing domains will have different representations, goals, methods of evaluations, plus further differences in relation to the algorithmic methods applied to the given problem. While it may be within the underlying concept that the creativity exists, this concept is generally wrapped up in layers of representation and computation specific to the given domain.

In examining the discussion of domains in relation to general and computational creativity, we find it difficult to prize creativity and application domain apart. According to (Plucker and Beghetto 2004), domain-dependency may not matter, yet when we consider a specific CC system it appears that it might. We have argued that these systems do not (and arguably can not until they acquire free-will) select the domain in which to work, this is generally chosen by the programmer. If the choice of domain is critically important or if CC is not independent in relation to domain, is this an argument against the idea that computational systems can be autonomously creative? If the application domain is chosen by the programmer we cannot separate the evaluation or impact of the CC system from its domain. This implies that reciprocally, we cannot separate the impact of a CC system from its programmer.

So can we answer the question 'Is CC domain general?'? In reality, a yes or no response to such a question trivialises that which is being asked. If we consider the field of CC, then yes all domains may be investigated but when we consider an individual system, more often than not we place all experimentation and evaluation into a given domain. The closer one looks at a given computational system, the more tied in and restricted to the boundaries of the application domain one becomes. This should not impede the development of any CC research, however. As long as the trend of considering new and diverse applications continues, then the scope of studies within the field and range of knowledge obtained through the field can only expand in the coming years. A strongly creative AI capable of general intelligence would arguably be domain-general, but until such a system exists, it appears that any implemented CC system is limited to the domain its user designed it to work in — which, of course, could be any domain at all.

Acknowledgments

We would like to thank the reviewers for their detailed and insightful comments. This work is part of the App'Ed (Applications of Evolutionary Design) project funded by Science Foundation Ireland under grant 13/IA/1850.

References

al Rifaie, M. M., and Bishop, M. 2015. Weak and strong computational creativity. In *Computational creativity research: Towards creative machines*. Springer. 37–49.

Amabile, T. M. 1995. Attributions of creativity: What are the consequences? *Creativity research journal* 8(4):423–426.

Ariza, C. 2009. The interrogator as critic: The turing test and the evaluation of generative music systems. *Computer Music Journal* 33(2):48–70.

Baer, J. 1998. The case for domain specificity of creativity. *Creativity research journal* 11(2):173–177.

Baer, J. 2010. Is creativity domain specific. *The Cambridge handbook of creativity* 321–341.

Besold, T. R. 2016. The unnoticed creativity revolutions: Bringing problem-solving back into computational creativity. In *Proceedings of AISB Third International Symposium on Computatational Creativity*.

Bhattacharjya, D. 2016. Preference models for creative artifacts and systems. In *Proceedings of the Seventh International Conference on Computational Creativity*.

Boden, M. A. 1998. Creativity and artificial intelligence. *Artificial Intelligence* 103(1):347–356.

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

Bown, O. 2012. Generative and adaptive creativity: A unified approach to creativity in nature, humans and machines. In *Computers and creativity*. Springer. 361–381.

Bringsjord, S.; Bello, P.; and Ferrucci, D. 2003. Creativity, the Turing test, and the (better) Lovelace test. In *The Turing Test*. Springer. 215–239.

Cardoso, A.; Veale, T.; and Wiggins, G. A. 2009. Converging on the divergent: The history (and future) of the international joint workshops in computational creativity. *AI Magazine* 30(3):15.

Colton, S.; Wiggins, G. A.; et al. 2012. Computational creativity: the final frontier? In *ECAI*, volume 12, 21–26.

Colton, S. 2008. Creativity versus the perception of creativity in computational systems. In *AAAI Spring Symposium: Creative Intelligent Systems*, 14–20.

Cook, M., and Colton, S. 2015. Generating code for expressing simple preferences: Moving on from hardcoding and randomness. In *Proceedings of the Sixth International Conference on Computational Creativity June*, 8.

Cropley, A. J. 2000. Defining and measuring creativity: Are creativity tests worth using? *Roeper review* 23(2):72–79.

Csikszentmihalyi, M. 2013. *Creativity: The Psychology of Discovery and Invention, reprint ed.* Harper Perennial.

Fauconnier, G., and Turner, M. 2003. Conceptual blending, form and meaning. *Recherches en communication* 19(19):57–86.

Guckelsberger, C.; Salge, C.; Saunders, R.; and Colton, S. 2016. Supportive and antagonistic behaviour in distributed computational creativity via coupled empowerment maximisation. In *Proceedings of the Seventh International Conference on Computational Creativity*.

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, Atlanta*.

Guilford, J. 1950. Creativity. *American Psychologist* 5:444–454.

Jordanous, A. 2011. Evaluating evaluation: Assessing progress in computational creativity research.

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.

Loughran, R., and O'Neill, M. 2016. Generative music evaluation: Why do we limit to 'human' ? In *Computer Simulation of Musical Creativity (CSMC)*.

Loughran, R., and O'Neill, M. 2017a. Application domains considered in computational creativity. In *Proceedings of the 8th International Conference on Computational Creativity, Atlanta*.

Loughran, R., and O'Neill, M. 2017b. Limitations from assumptions in generative music evaluation. *Journal of Creative Music Systems* 2(1).

Marsden, A. 2013. *Music, intelligence and artificiality.* Routledge. 15–29.

Meusburger, P. 2009. Milieus of creativity: The role of places, environments, and spatial contexts. In *Milieus of Creativity*. Springer. 97–153.

Nebel, C. 1988. *The dark side of creativity: Blocks, unfinished works and the urge to destroy*. Whitston Publishing Company.

O'Hara, M., and Auersperg, A. M. 2017. Object play in parrots and corvids. *Current Opinion in Behavioral Sciences* 16:119–125.

Plucker, J. A., and Beghetto, R. A. 2004. Why creativity is domain general, why it looks domain specific, and why the distinction does not matter. *Creativity: From potential to realization.*

Plucker, J. A. 2004. Generalization of creativity across domains: Examination of the method effect hypothesis. *The Journal of Creative Behavior* 38(1):1–12.

Ritchie, G. 2001. Assessing creativity. In *Proc. of AISB'01 Symposium*. Citeseer.

Ritchie, G. 2006. The transformational creativity hypothesis. *New Generation Computing* 24(3):241–266.

Ritchie, G. 2007. Some empirical criteria for attributing creativity to a computer program. *Minds and Machines* 17(1):67–99.

Runco, M. A., and Acar, S. 2012. Divergent thinking as an indicator of creative potential. *Creativity Research Journal* 24(1):66–75.

Runco, M. A., and Jaeger, G. J. 2012. The standard definition of creativity. *Creativity Research Journal* 24(1):92–96.

Runco, M. A. 1987. The generality of creative performance in gifted and nongifted children. *Gifted Child Quarterly* 31(3):121–125.

Runco, M. A. 2014. *Creativity: Theories and themes: Research, development, and practice*. Elsevier.

Saunders, R., and Bown, O. 2015. Computational social creativity. *Artificial life* 21(3):366–378.

Stein, M. I. 1953. Creativity and culture. *The journal of psychology* 36(2):311–322.

Still, A., and d'Inverno, M. 2016. A history of creativity for future ai research. In *Title: Proceedings of the 7th Computational Creativity Conference (ICCC 2016). Universite Pierre et Marie Curie*.

Urban, K. K., and Jellen, H. G. 2005. *Test of Creative Thinking: Drawing Production: TCP-DP*. Harcourt Assessment, Incorporated.

Ventura, D. 2016. Mere generation: Essential barometer or dated concept. In *Proceedings of the Seventh International Conference on Computational Creativity, ICCC*.

Ventura, D. 2017. How to build a CC system. In *Proceedings of the 8th International Conference on Computational Creativity, Atlanta*.

Wiggins, G. A. 2006. Searching for computational creativity. *New Generation Computing* 24(3):209–222.

Wiggins, G. A. 2012. The mind's chorus: creativity before consciousness. *Cognitive Computation* 4(3):306–319.

y Pérez, R. P. 2015. From mexica to mexica-impro: The evolution of a computer model for plot generation. In *Computational Creativity Research: Towards Creative Machines*. Springer. 267–284.

A Massive Sarcastic Robot: What a Great Idea!

Two Approaches to the Computational Generation of Irony

Tony Veale
School of Computer Science and Informatics
University College Dublin, Belfield D4, Ireland.
Tony.Veale@UCD.ie

Abstract

Irony is a versatile seasoning for language that is just as useful for sugaring insults as for salting compliments. It is a flavoring that adds bite to much of our online interaction, making its computational analysis – recognition *and* understanding – a necessity for affective language processing. If the computational *generation* of irony is a harder sell, it is perhaps because this mode of communication is so often vexing for humans too. However, an artificial fluency with irony is as desirable as fluency in *any* form of creative language, from metaphor and analogy to humour and persuasive argumentation. We explore two distinct approaches to irony generation in this paper: knowledge-based generation *ab initio*, and a more shallow approach we name 'mere *re*-generation.' We consider the relative merits of each, conduct a user evaluation, and demonstrate some practical uses.

The Devil's Seasoning

To communicate with irony is to talk with a forked tongue. Yet while the need for the computational analysis of irony is strong, since so much of our online language is playfully or scornfully sardonic (Hao & Veale, 2010; Reyes *et al*., 2013; Ghosh & Veale, 2017), the case for computationally *generating* irony is much less compelling. However, there are practical reasons for granting our machines a fluency in this challenging mode of communication. Irony requires a delicate blending of mental spaces to concisely express a double-edged attitude to a failed expectation: we highlight an expectation, act as though it were fulfilled, and criticize its lack of fulfilment, all in a single breath. So a generative model of irony is also, by necessity, a model of conceptual blending (Fauconnier & Turner, 2002), one that lends not just concision, but creativity too, to a machine's outputs.

An ironic statement can be a most charming disguise for a conceptual conflict, though the charm of the disguise and the profundity of the conflict will vary from one speaker to another – skill is a factor, after all – and with the obviousness of the context. Irony is a subtle form of sarcasm (conversely, sarcasm is a vulgarized form of irony) in which we believe in what we say but not *how* we say it. So when we use the norms of flattery to sweeten an insult, or the norms of criticism to sour a compliment, we intend an audience to appreciate the seasoning but to look beyond it too, to grasp our deeper meaning and our ambivalent attitude towards it. Ironies, like metaphors, are allusive (if sometimes *elusive*) products of the imagination that appeal to the imaginations of others. But these products are not built in a vacuum; an ironic worldview makes a critical claim about something in the world that others can see and evaluate for themselves. For a machine to generate an ironic observation, it needs knowledge of how things are and how they should be, and it needs the ability to frame the gap between the two in a pithily suggestive fashion. So, as befits the duality of irony, we explore *two* approaches to its generation in this paper.

The first is a knowledge-based approach that explicitly models that which is expected and that which is observed, so as to juxtapose them both in the same tweet-sized text. We use *dis*analogy as the unifying conceit to maximize the dissonance between the two perspectives, but the resulting text also supports other markup strategies to signal ironic intent to an audience. While the most obvious applications of machine-generated irony may well be human-computer interfaces, a less obvious, but no less useful, application is the generation of well-controlled test-data for experiments into the human appreciation of irony. Our evaluation of this parameterized approach shows how linguists can better understand how we as humans process irony, by presenting human subjects with the products of machine creativity.

How do we humans acquire our sense of the ironic? This sense is not something we are born with, but something we cultivate over time, such as via continuous exposure to the ironic stylings of others. A machine can also be exposed to ironic language to learn its signature qualities, with social media platforms such as Twitter making it easy to retrieve large amounts of texts that are self-annotated with the hash tags *#irony*, *#sarcasm* and *#yeahright*. The overt tagging of verbal pretence is not a new phenomenon, even if Twitter elevates the practice to new levels of explicitness. Speakers have always used subtle cues to signal their ironic intent. By harvesting a broad spectrum of cued ironic texts from the web, a machine can build a large case-base of attested examples to be reused wholesale, or recycled with novel variations, as parts of its own future utterances. Our second approach does not analyse these web examples in any great depth, but pursues a philosophy we dub *mere regeneration* to find new uses and framings for old word combinations. The qualities that make one juxtaposition of words or ideas seem more poetic, more beautiful, more ridiculous or more hilarious than others may not always defy logical scrutiny, but for all practical intents they remain ineffable for now. Our machines should thus do what most humans do at one time or another: reuse the resonant combinations of words that have worked for others and claim them for themselves.

We present each of these approaches in the sections to follow, beginning with a review of related work and ideas in the next. After an empirical evaluation of the first, and a discussion on how the second supports the rapid development of humorous CC systems, the paper concludes with a discussion of the relative merits of each approach to irony.

Related Work and Ideas

To judge by the diversity of tweets that Twitter users tag with *#irony*, the general public operates with a somewhat diffuse understanding of what irony is. The popular view, and the most oversimplified, is that to speak ironically is to say one thing but to mean its opposite (Kierkegaard, 1841; Grice, 1978). But irony is a nuanced idea that demands just as nuanced a definition, and *irony-as-opposite* disappoints on several fronts: it is meaningful for just a subset of the utterances that speakers intuitively grasp as ironic; even in such cases, opposition accounts for just one aspect of the intended meaning; and even then, it is not always obvious how one can arrive at the opposite of an ironic statement. Recall a scene in the film *Amadeus* in which the composer Salieri has just premiered his new opera. When put on the spot for a positive response, Mozart shrewdly replies that "*When one hears such music, one can only think 'Salieri'!*" Mozart's words are criticism masked as flattery and Salieri suspects as much, though he cannot know for sure. But what is the opposite of Mozart's reply here – that one does *not* think of Salieri when hearing such music? No, what is inverted here is not what Mozart says but what he implies, "When one hears such [*lovely*] music, one can only think 'Salieri'!" His ironic meaning thus becomes "When one hears such [*unlovely*] music, one can only think 'Salieri'!"

To speak ironically then is to say one thing, insincerely imply the obvious, and intend something so different that it often amounts to the opposite of what is implied. This constitutes an act of *verbal pretence* (Clark & Gerrig, 1984) and *pragmatic insincerity* (Kumon-Nakamura *et al.*, 1995) that is designed to be penetrated by audiences. But if ironic statements are meant to be understood as such, they spark a conflict of implications between the default and the non-default (Giora *et al.,* 2015), or the obvious and the creative, that audiences must somehow resolve for themselves. The context often determines how fraught this conflict will be, with contexts that are strongly supportive of an ironic interpretation nudging audiences to look past the obvious. In contexts that are equally supportive of the default and non-default interpretations, the audience is left – like Salieri – in a rather uncomfortable superposition of affective states.

In such cases, authors have a number of ways to nudge an audience toward the creative. Many ironic utterances are context-*external*, which is to say that the information one needs to discern ironic from non-ironic is found *outside* the utterance itself. Mozart's reply is a context-external irony. Many more are context-*internal*, insofar as an author bakes the necessary context into the utterance itself. For example, consider this classic image from *Farewell My Lovely* by Raymond Chandler: "He looked about as inconspicuous as a tarantula on a slice of angel food cake." The *He* is Moose Malloy, a hulking white brute who is newly-released from prison and in search of his faithless wife Velma in a black neighborhood of the city. Throwing discretion to the wind, the white Malloy stomps about town, terrifying the locals while sticking out like a large black spider on a white cake. The comparison is enough to alert readers that Malloy is the very opposite of inconspicuous. Yet note that the simile means more than "Malloy was very conspicuous indeed." It means "Malloy should have tried to be inconspicuous, but the dumb brute could not be subtle if he tried." We use irony to conflate perspectives *and* to criticize, all at once.

Besides building useful context into his simile, Chandler also prefaces the comparison with "about", a marker of imprecision that alerts readers to his insincere use of words. Hao & Veale (2010) use these markers of imprecision to harvest creative similes in bulk from the web. While the marker "about" is not reserved for ironic comparisons – that depends on whether the simile is intended as flattery or criticism – it *is* a reliable marker of linguistic creativity. As analyzed in Veale (2013), about-similes tend to use longer descriptions (or "vehicles") that constitute what Fishelov (1992) deems PS (poetic similes). He contrasts these with the NS (non-poetic) similes that pervade language, such as "clear as mud", "light as a feather" and "dry as a bone." PS similes use many of the same words as NS similes, but use them in striking juxtapositions that are memorable and sometimes hilarious, as in "as sophisticated as a zombie at a dinner party" and "as quiet as a cat in a blender." Taylor (1954) laboriously compiled a large corpus of PS similes that had become the stuff of proverb in California – such as "as useful as teats on a boar" – but markers such as "about" allow our machines to amass such corpora automatically.

Despite helpful markers such as "about", "almost" and "not exactly", irony tends much less to the formulaic than sarcasm. The latter does not sustain its verbal pretence for very long, nor does it leave its audience in much doubt as to the true intentions of a speaker. Sarcastic tweets such as the following are thus a commonplace on Twitter: "I love it when my 'friends' forget my birthday" and "Don't you just love it when your boss throws you under the bus?" Without the safety-net of face-to-face interaction, Twitter users are careful to signal their insincerity openly, as misunderstandings on social media can lead to public shaming. The fear of public rebuke is so strong that users routinely tag even the most formulaic sarcastic tweets with *#sarcasm*. So it is an easy matter to harvest large amounts of apt training data from Twitter, to train our machines to recognize a sarcastic attitude using supervised machine-learning techniques.

Statistical classifiers can use everything from the words themselves to their POS tags, bigram / trigram collocations and sentiment scores to discriminate sarcastic from non-sarcastic texts. Riloff *et al.* (2013) used the mixed emotions of sarcasm as a characteristic signature, and obtain good results on the short texts that are typical of Twitter. Reyes *at al*. (2013) used a broader basket of features, including symmetry, to identify both irony and wit more generally. Ghosh *et al*. (2013) focused not on irony or sarcasm detection but on the estimation of sentiment in figurative tweets that comprise ironic, sarcastic and metaphorical examples; their annotated corpus is frequently used as a training and test set for sarcasm detection. Ghosh & Veale (2015) first trained a neural network to recognize sarcasm, and later (Ghosh & Veale 2017) extended this network to integrate a model of sarcastic mood. When working with tweets, a machine has access to a timestamp for each, to author data, and to the timeline in which each was posted. Using the web service *AnalyzeWords* (Tausczik & Pennebaker, 2010) to perform a mood analysis of the prior 100 posts leading up to a given tweet, an extra 11 dimensions – including *anger*, *positivity*, *remoteness*, *worry* and *analyticity* – can be added as network inputs. Ghosh & Veale show that this personal context is as useful as the usage context of a tweet (i.e. the text to which it was posted in reply) in recognizing the user's pragmatic intent. Those authors also introduced another innovation to the detection of sarcasm: rather than use independent raters to annotate the training and test sets for sarcasm, they used a Twitterbot, *@SarcasmMagnet*, to contact the owner of each tweet directly, to obtain in real time the author's own statement of pragmatic intent.

These approaches still bring with them a concern about over-fitting. Are the features that prove to be most useful at detecting sarcasm and irony truly generic, or do they just happen to be the words that best separate the positive from the negative instances in a particular testset? These systems perform detection without ever striving for understanding, but we humans take a very different approach: to recognize an act of pragmatic insincerity, we first analyse its intent in terms of the meaning it might communicate to others. This analysis is crucial for the generation of irony, for a system cannot be ironic if it does not know what it intends to say or cannot know if it has faithfully conveyed that intention. Our best statistical models of detection are too shallow to be reversed to serve as models of generation, so we are still quite some way from a CC system that can accept "Your music sucks, Salieri!" as an input and generate as its output "When one hears such music, one can only think 'Salieri'!" With this dour prognosis in mind, we limit ourselves in the next section to a highly-structured expression of irony that machines can both generate and appreciate for themselves.

EPIC Fails

To give a machine a capacity for generating irony, we must first break its heart. For whatever else irony might be, and regardless of whether it is used to criticize (its main use) or to praise (a minority pastime), every ironic statement is an expression of disappointment. A machine whose sole job is to be ironic is a machine that must always be disappointed. Since disappointment results from a failed expectation, our ironic machine must thus possess a model of expectation.

We propose a simple model of property-oriented expectation, named EPIC, in which an expectation (E) predicts a property (P) of an instance (I) of a concept (C). Take the concept of a "party." An instance I – my birthday party, say – of this concept C carries with it one or more expectations (E) of the typical properties (P) of parties: so we expect I to be fun, to be entertaining and to be social. An expectation E fails if the expected property P cannot be asserted of I, and fails ostentatiously if we can instead assert its opposite, not-P. Even if E fails in a more subtle way, the task of the ironist is to exaggerate the truth for humour's sake. In this way, a failed expectation E_1 of I_1 concerning P can match an expectation E_2 of a non-salient concept C_2 that predicts not-P. So just as parties should to be fun and entertaining, we often expect lectures to be dull and boring. In failing to be fun, a party I_1 fulfils an expectation of lectures that few guests actually bring to a party. But by matching a failed expectation for P to an non-salient expectation for not-P, an ironist can dramatize the non-P of I_1 (an instance of C_1) by pretending that I_1 is an instance of C_2 that entails the expectation not-P, or can perhaps feign mock surprise to have attended I_2 (an instance of C_2) instead of I_1.

Even the most committed ironists spend only a tiny part of their lives being ironic. The rest of their time is dedicated to the stuff of everyday life: working, reading, shopping and interacting with others. A machine whose sole task is to be ironic is an oddity indeed, but how is it to acquire the expectations that we humans spend a lifetime developing? The answer, as in many NLP tasks, is the web. To acquire the expectations E that people bring to instances I of the concepts C, a machine can consider the adjectives "P" that adorn the word "C" in common usage. The Google ngrams database (Brants & Franz, 2006) provides a large inventory of frequent web collocations. Consider these 3-grams:

W1	W2	W3	*Web Count*
a	fun	party	*10060*
a	dull	party	*772*
an	entertaining	party	*161*
a	boring	lecture	*1882*
a	dull	lecture	*267*

Notice how "dull" is more used often, in absolute terms, to describe parties than lectures, so frequency alone is not a reliable indicator of expectation strength. Machines can use n-gram data to suggest apt candidates for property-oriented expectations, but they must look elsewhere to confirm their hypotheses. Since similes are linguistic constructions that take full advantage of conceptual expectations, a machine can determine whether P is a widely-held expectation for instances of C by looking for similes of the form "as P as a C." Veale (2012) shows how expectations that conform to the EPIC structure are harvested in bulk from the web by retrieving all matches for the wildcard query "as * as *."

Our machine shall also need some relational knowledge, to understand how others typically relate to the concepts C that are so-described with a property P. For instance, how do people relate to parties or lectures, and can we relate to instances of each in the same way? If so, an analogy can be constructed from the shared relationships. A relation is any triple <C_2 R C_1> linking two concepts in the abstract and two instances of those concepts in the specific. The query logs of web search engines are a good source of common-sense triples, since users expose their expectations of the world in the questions that they pose online. So when a pet owner asks "why do dogs chase cars" or "why do cats arch their backs" these questions assume that everyone else believes that <dogs chase cars> and <cats arch backs> too. Veale & Li (2011b) show how a large database of question-derived triples can be "milked" from the query continuations offered by Google. While its query log is private, when the engine suggests popular completions for partial queries it is effectively exposing recurring entries on that log.

The thwarted expectation in which an ironic utterance in rooted can take many forms. EPIC assumes that the expectation concerns a property P for concept C_1, but it can be extended to a concept C_2 by the relation <C_2 R C_1>. To highlight a failure to observe P of C_1, an ironist can compare C_1 to a C_3 for which not-P is expected, on the basis of a parallel relation <C_4 R C_3> and the analogy C_1:C_2::C_3:C_4. Since C_1 and C_3 are not so much compared as contrasted on the basis of a conflict between P and not-P, the juxtaposition is more disanalogy than analogy. In our example of parties and lectures. the disappointment of a failed event can be conveyed with irony with the following disanalogy:

Some hosts arrange "entertaining" parties the way presenters arrange boring lectures.

We can now appreciate the function of the shared relation R (in this case, *arrange*): it focuses the ironic charge of the disanalogy toward those who arrange the parties that fall so short of our expectations, in the same way that explosives experts shape their charges to explode in a given direction. By wrapping the expected property "entertaining" in ostentatious scare-quotes, the charge appears to be echoing a lie, a failed prediction that a speaker now mimics with ridicule. The "echoic mention" of an unwise prediction (Sperber & Wilson, 1981; Kreuz & Glucksberg, 1989) offers a way of elevating the veiled criticism of irony into open mockery. If the criticism were expressed on Twitter, a speaker might go so far as to append the hashtag *#irony*, as if to say "Isn't it ironic when …" The relative merits of these strategies – disanalogy, scare-quotes and overt tagging – for conveying an ironic worldview will be evaluated in the next section.

EPIC Succeeds

The success of an ironic utterance hinges on its capacity to highlight the failure of a reasonable expectation. As some are more successful in this regard than others, we need a gradated yardstick of success that goes beyond the binary. Notice that while EPIC predicates success on the inference of not-P in a context that implies P, it does not subscribe to an unnuanced irony-as-opposition view. Instead, it assumes that irony is successful when audiences shift their expectations of C from P toward not-P either in whole or in part. A successful ironic utterance may leave audiences with the mixed feeling that instances of C occupy a middle-ground between P and not-P that conforms to neither extreme; for example, that "many parties that promise entertainment are only ever entertaining to the people that host them." While we cannot measure nuanced feelings like this, Valitutti & Veale (2017) propose a convenient proxy: if P is a positive property and not-P is a negative property, then an ironic statement in the EPIC mold is successful to the extent that audiences downshift their mean rating of P's positivity in the context of the irony. We can expect, for instance, that the mean positivity of the property "entertaining" in a null context is higher than its mean rating in the context of a disanalogy that lends the word a halo of disappointment.

This gradated downshifting view permits us to measure success for irony generation overall, as well as the relative contribution of our different strategies – disanalogy, scare-quotes and overt tagging – to this success. We conduct a crowd-sourced evaluation using the platform *CrowdFlower* in which anonymous judges are each paid a small sum to rate the positivity of focal words in the ironic utterances constructed using EPIC. The focal word in each case is the property P, or in other words the adjective that is placed in scare-quotes. We use our generative system to generate 80 distinct ironic utterances, with the same structure as our party/lecture example, around a different focal property in each case. Each test instance exploits an expectation E for a positive property P that, we expect, is shifted toward a negative evaluation by the use of a disanalogy with another expectation E' for not-P. Here is one such test instance:

#irony: *When "cultured" gentlemen pursue ladies the way feral predators pursue prey*.

This is the fully-loaded version of the output, including the disanalogy, the scare-quotes and the overt tag. A number of other variants can be generated by ablating one or more features, and by asking judges to rate alternate variants of the same observation about a focal property, we can tease out the relative impact of each feature to the downshift. We label each variant as shown in the following examples:

BASE:
Cultured gentlemen pursue ladies

BASE+QUOTE:
"*Cultured*" *gentlemen pursue ladies*

BASE+COMP (disanalogy)
Cultured gentlemen pursue ladies the way feral predators pursue prey

BASE+QUOTE+COMP:
"*Cultured*" *gentlemen pursue ladies the way feral predators pursue prey*

BASE+QUOTE+COMP+HASH:
#Irony: "*cultured*" *gentlemen pursue ladies the way feral predators pursue prey*

We provide alternate variants of the same utterance to different judges, and ask each to estimate the positivity of the focal word on a scale from +1.0 (most positive) to -1.0 (most negative). We elicit ten ratings per utterance variant and then calculate the mean positivity rating for each. But to appreciate the extent of the ironic shift, we need to know how judges would rate these focal words in a null context, free of the baleful influence of the ironic utterance.

In another CrowdFlower experiment, one that is actually conducted prior to the one above, we do precisely this. We provide the 80 focal properties from the 80 automatically-generated utterances – words such as "entertaining" and "civilized" and "smart" and "creative" – and ask judges to rate their overall positivity on the same +1.0 to -1.0 scale. The mean ratings provide an estimate of the positivity of the words in their primary dictionary senses. We can now calculate the mean shift in positivity caused by an ironic utterance; the means are displayed in Table 1 below, with standard deviations in parentheses.

Structural Variant	*Mean Positivity*
BASE	0.51 (*SD* 0.38)
BASE+QUOTE	0.41 (*SD* 0.46)
BASE+COMP	0.29 (*SD* 0.49)
BASE+QUOTE+COMP	0.20 (*SD* 0.54)

Table 1. *Mean positivity of the focal words in ironic utterances with different structural variants. All differences between conditions are significant at the p < .001 level.*

As shown in Table 1, each successive feature increases the mean downshift in perceived positivity of a focal word P and its associated expectation E, with disanalogy offering the most forceful shift into negative territory. We can ask how often an utterance succeeds in not just diminishing the positivity of a focal word but in making it appear negative to an audience. Table 2 reports how likely the focal word is to be seen as positive overall by raters.

Structural Variant	*Positive Likelihood*
BASE	0.91 (*SD* 0.15)
BASE+QUOTE	0.82 (*SD* 0.13)
BASE+COMP	0.75 (*SD* 0.15)
BASE+QUOTE+COMP	0.64 (*SD* 0.16)

Table 2. *Likelihood that a focal word is viewed as positive rather than negative in different structural conditions.*

Again these show that disanalogy has a greater impact than scare-quotes on the upending of perceived sentiment, while combining *both* features yields a larger impact still. But the experiments also point to a negative finding not shown in Tables 1 and 2: overt marking with *#irony* has no discernible impact on utterances that already use scare-quotes and disanalogy, and has far less impact than either of those variants when it is used without them. It is one thing to explicitly announce an ironic mindset, and quite another to seed it effectively (and affectively) in the minds of an audience.

Mere Re-Generation

These tightly-organized utterances add nuance to the irony-as-opposition debate by effectively creating an ambivalent middle ground between an expected property P of C and its negation, not-P. But they do this by appealing to the experience of an audience rather than to its imagination. It takes experience – of parties and lectures, for example – to appreciate how our expectations can be fulfilled or thwarted. But no strange new concepts are introduced in these utterances, and no category boundaries are challenged. Instead, concepts are used in their simplest guise. In contrast, the "about" similes harvested by Hao & Veale (2010) and analyzed in Veale (2013) offer a surfeit of vivid detail to help us visualize a concept. In these similes we are told not just of parties but of *grunge* parties, *frat* parties, *hen* parties, *stag* parties, *beach* parties and *tea* parties. These events attract equally vivid guests: *wasps* at a tea party, a *skunk* at a lawn party, a *zombie* at a dinner party, a *teetotaller* at a frat party and even, absurdly, a *jackboot* at a testicle party. We also hear not just of lectures, but of six hour lectures on mahogany, advanced lectures on theoretical physics by Stephen Hawking and 40-minute lectures on pockets! The humour resides as much in the detail of C_1 as it does in any juxtaposition between C_1 and another concept C_2. Or, rather, the concept C_1 is already a vivid mix of ideas.

Veale (2011a) presented a system, the *Jigsaw Bard*, that repurposes n-gram collocations as descriptive vehicles for novel similes. For instance, the Google 2-gram "robot fish" names a family of aquatic drones, but, as evidenced by the stock similes "as cold as a fish" and "as cold as a robot," it might also describe a person who is emotionally cold. Our words carry a myriad unspoken constraints that are grasped only by fluent speakers, so the *Bard* sidesteps the challenges of building its imaginative word combinations *ab initio*. Rather, it uses a simple rule for locating its *objets trouvés* in web ngrams: a bigram "W_1 W_2" suggests how a concept combination C_1: C_2 for which the system already possesses the stock similes "as P as a W_1" and "as P as a W_2" can be repurposed as the novel simile "as P as a W_1 W_2." Shared expectations of P thus yield unified similes for P. Now, W_1 and W_2 are part of the *Bard*'s creative vocabulary by virtue of already serving as vehicles in its library of stock similes. But what if we reuse the vehicles from our *about* similes, which tend to be longer and more vivid, in the same way?

Many of those vehicles are inherently ridiculous. Just as irony is more than mere opposition, the ridiculous is more than mere absurdity. It occupies a place between the absurd and the impossible where our reaction is one of laughter or horror rather than puzzlement or stupefaction. It marks out another possible world that is out of joint with this one. So the similes of our *about* corpus speak of a dog in a sweater (insightful), pants in a nudist colony (necessary), a nun at a Reggae festival (inconspicuous) or a 10-ton rock in a canoe (useful). Each is the product of a personal sense of humour that can be as tasteless as leopard skin pants at a funeral or as welcome as a fart in a spacesuit. Each composition is a vivid conceptual blend (Fauconnier & Turner, 2002) that unites disparate ideas to spark emergent inferences. Since a dog in a sweater offers a surface imitation of human intelligence, the blend ironically undercuts – with some inference – a *pseudo*-intellectual who merely dresses for the part.

By harvesting a large corpus of *about* similes and their ridiculous mental images from the web, we can provide our machine with the rudiments of a composite sense of humour, a singular comedic voice formed out of the multitude. This corpus of ironic blends is also a comprehensive database of EPIC fails, which is to say, failures of expectations about properties that are vividly painted on a grand scale. We may try to dissect these failures into their individual parts, to take the dog out of its sweater and the nun out of her Reggae festival, so that an ironic machine can recombine the parts in new ways; perhaps by putting the nun in the spacesuit, the dog in the festival and the fart in the sweater. But the unspoken logic of the ridiculous that dictates how irony and humour emerge is unlikely to carry across to the new combinations. Those leopard skin pants may not seem so tasteless at a Reggae festival, nor a dog so conspicuous, and it is the air seal on a full-body spacesuit that makes the smell so much more unwelcome there than in a sweater. As Veale (2015) argues, jokes are compressed thought experiments, and humorous blends such as these can rely just as much on our physical intuitions as a conundrum in physics. We can no more chop up these blends and recombine their parts to generate a new one that is just as witty than we can chop up a science text to propose new theories in science.

The language may be robust at times, but the humour of these EPIC fails is fragile indeed. If an ironic machine is to exploit it reliably, it must make as few changes to possible. So the simplest and most reliable strategy is to reuse each blend in its entirety. Suppose we want to ridicule the lack of insight of a scientist or a reporter or of any kind of critic that speaks with authority. Our machine can retrieve from its database a blend that is ironically associated with the property *insightful*, such as our dog in a sweater, a plastic cap, a shaving foam commercial, a myopic mole in a sack, a college freshman essay, gravel, a rock, a fortune cookie, or a child writing home from summer camp, and attach this blend to our target. This mere *re*-generation is more effective than it is creative, yet it enables the rapid construction of CC systems like chatbots and interactive Twitterbots. Consider the political satire bot *@TrumpScuttleBot*, which offers a knowledge-based parody of a person whose tweets regularly flirt with the vulgar and the ridiculous. Rather than slice and dice the man's own tweets into a statistical gumbo of prejudice and provocation, as done by bots such as *@DeepDrumpf*, our parody bot works from first principles to create analogies and metaphors and whimsical political comparisons. For example, the bot often compares itself to other leaders, using EPIC to suggest the kinds of people that we expect to show leadership, as in this tweet:

> I come from a long line of HUGE demagogues who led a long line of TREMENDOUS mobs. The lying media don't tell you THAT! #MAGA

The satirical humour of the bot goes hand in hand with its own ineptitude, which is gleefully projected onto its target. Demagogues are leaders, yes, but not good leaders, and it is poor manners indeed to paint one's voters as a mob. But the bot also paints savage caricatures of its stock enemies:

> Those REGRETTABLE NBC hacks are as friendly as a rabid pit bull challenging all comers to a choice cut of steak! #MAGA

As the bot does not expect enemies to be *friendly*, it pulls out an ironic *about* simile. Other expectations are packaged as self-promotion, to highlight some key recurring themes:

> TRUMP University will re-open to offer TERRIFIC Degrees in Virtue to AMERICAN philosophers. Foreign philosophers are about as ethical as the tobacco industry! #MAGA

Echoing its target's distinctive language patterns, the bot makes frequent use of generic descriptors such as HUGE, TERRIFIC, SAD, and so on. When it talks about a specific concept C, such as *philosopher*, it can also bring specific EPIC expectations to bear. As the philosophers who study virtue are expected to be ethical, it looks to its database of blends to suggest an ironic comparison with Big Tobacco. The mere-regeneration of ironic similes here plays second-fiddle to a main act that revolves largely around analogy.

Mere re-generation is especially useful in deciding how a bot should respond to the tweets addressed to it by other, human users on Twitter. *@TrumpScuttleBot* has three strategies for replying to humans: in the first, it matches words in the human's tweet to the entries in a dictionary of dream symbols, rewriting the text of the interpretation in Twitter terms so that "to dream about" becomes "to tweet about"; in the second, it identifies a property P in the human tweet for which it can find an ironic simile in its database; and in the third, it thanks the human for a comment that is one of *{topical, useful, fascinating, deep, insightful, intelligent}*, and attaches an aptly ironic simile to subvert this gratitude. Here is a real exchange between the bot and a human user:

From @Anonymized, replying to @TrumpScuttleBot
Ties? He couldn't be further away.

From @TrumpScuttleBot, replying to @Anonymized
Thanks for the useful feedback. It's as useful as a bag full of farts.

From @Anonymized, replying to @TrumpScuttleBot
Open it up and get a whiff … aaaaahhhhhhhh

From @TrumpScuttleBot, replying to @Anonymized
To tweet of being or moving up suggests you are emerging from a depressing or negative situation.

From @Anonymized, replying to @TrumpScuttleBot
Yes Dr. Wizard

From @TrumpScuttleBot, replying to @Anonymized
To mention a wizard in your tweet suggests you are trying to hone your skills and exercize your power

From @Anonymized, replying to @TrumpScuttleBot
OK Goofy, if it makes you feel better.

From @TrumpScuttleBot, replying to @Anonymized
You make Jar Jar Binks seem about as goofy as doomsday

From @Anonymized, replying to @TrumpScuttleBot
Good one, you got me.

From @TrumpScuttleBot, replying to @Anonymized
You make others seem about as good as a movie about a leprechaun in space

From @Anonymized, replying to @TrumpScuttleBot
At least Trump is president.

From @TrumpScuttleBot, replying to @Anonymized
Thanks for the useful feedback. It is as useful as a knock on the head.

From @Anonymized, replying to @TrumpScuttleBot
You keep responding, so who's head is being knocked?

From @TrumpScuttleBot, replying to @Anonymized
To mention a head in your tweet signifies wisdom, intellect, understanding and rationality.

A rapid-response bot like *@TrumpScuttleBot* always gets the last word, but the creative challenge is in sustaining an engaging, if combative, banter with sporadic flashes of wit. Notice how the bot enriches a blend from its database with an extra flourish of its own. If a tweet contains a property P associated with an entity, fictional or real, in its database of familiar faces, the bot mixes that character into the blend too, as when it compares a "goofy" user to *Jar Jar Binks*.

Comedy Gold

Our machine should not meddle with the distinctive mix of images in a prebaked blend, but it might rework its syntax. Consider our ironic bot, *@OldSkoolFunBot*, which aims to generate witty banter by repackaging the ironic blends in its database of *about* similes, as in the following tweets:

> Question: Where are you most likely to find a wine stain? Well, in my world, how about on a white shirt?

> If you're like me you'll absolutely despise cooking spaghetti in the washing machine – What's that all about?

> Kids nowadays have their iTunes but in MY day we had to make do with being strapped to the rack

> I gave the mother-in-law the south end of a north-bound spiny lobster for Valentine's day but the grouch said my gift wasn't GOOD-LOOKING enough

The bot derives its sense of the ridiculous from the ways in which images are juxtaposed in ironic *about* similes. These juxtapositions are transformed via mere re-generation into quips that are just as ridiculous, even if no longer similes. New comic forms for prebaked combinations can be added quickly to adapt the bot to a new social trend. Consider the world of microbrew gastropubs, craft beers, and the quirky names that draw trend-setters to them. English pub names are famous for their naming conventions, in pairings such as "The Duke and Pony" and "The White Hart." To generate novelty pub names, *@OldSkoolFunBot* ekes out a pair of juxtaposed images from an *about* simile, to invent new names such as "The Porkchop and Synagogue", "The Dog and Sweater" and "The Fart and Spacesuit." To invent eye-catching new brands of craft beer, it reframes similes such as "friendly as a rabid dog" and "firm as a wobbly jelly" as "Rabid Dog IPA" and "Wobbly Jelly ale," in the hope that the humour of the juxtapositions persists in the new syntax. Here are some sample tweets from the bot in this vein:

> I'm off down to my local microbrew pub, The Drop And Bucket, for a pint of Mediaeval Ordeal ale.

> Fancy going down to the new microbrewery, The Bug And Rug, for a pint of Hungry Snake weizenbeer?

> I'm off down to my local microbrew pub, The Elephant And Tutu, for a pint of Triple Espresso lager.

These confections are more than random but less than fully appreciated by the bot itself. However, the random aspect does allow for unplanned resonances to emerge, as in:

> Fancy going down to the new microbrewery, The Dog And Wheelbarrow, for a pint of Golden Retriever lager?

> Fancy going down to the new microbrewery, The Fart And Car, for a pint of Ford Corsair ale?

Many other re-generation opportunities present themselves for rapid development in this way, such as in the naming of movie sequels by *@InterCableBot* or the naming of books by *@BotOnBotAction*. Re-generation bots like these aim to invent something new that has the magic of something old.

Stranger Then Friction

On encountering a robot with a humour setting, Cooper, an astronaut on a daring mission in the 2014 film *Interstellar*, says: "A massive, sarcastic robot. What a great idea." The robot in question, TARS, is blessed with a sense of humour that encompasses the sarcastic and the ironic, and proves to be a most excellent partner in the execution of the mission. KIPP and CASE, the film's other two robots, are said to be newer and faster than TARS, yet their senses of humour are less developed, presumably because TARS has more of what it takes to be witty: experience of people in the world.

Ironists use this experience to play one kind of friction against another: to exploit a friction between ideas, and the gap between expectation and reality, to lessen the tension between two people or between people and a machine. An ironic machine is attuned to disappointment yet knows how to repackage failure as amusement. Two approaches to this transformation have been presented here: a tightly controlled form of disanalogy that conflates an expectation and its failure in a single affect-shifting utterance, and a form of creative quotation that reuses attested examples of irony in new descriptive contexts that make them relevant again. Each approach can be modulated at the level of presentation to achieve effects that are more pointed or more subtle, but each works with different materials. In comedic terms, the first is a *straight* man that uses propositions that are not humorous in themselves to explain a failure of reasonable expectations; the second is a *funny* man whose material resonates with our own negative experiences of the world, but who encourages us to laugh at those experiences.

As with any successful comedy partnership, the perfect ironist is a marriage of both approaches, in which the first brings buttoned-down control and the second brings manic energy. This tight integration of structure and imagination has yet to be achieved in a single computational generator, even if each approach can be implemented side-by-side in a single system, such as a Twitterbot. Generative grammars (in a format named *Tracery*; see Compton *et al*., 2015) for the bots presented in this paper are available for download from *github.com/prosecconetwork*. This forced marriage of approaches will remain a chaste two-bed affair until we can better appreciate the magic of a screwball juxtaposition in computational terms. Until then, that "massive, sarcastic robot" will remain "a great idea," but a fictional one too.

References

Brants, T. & Franz, A. (2006). Web 1T 5-gram Version 1. *Linguistic Data Consortium*.

Clark, H. H., & Gerrig, R. J. (1984). On the pretense theory of irony. *J. of Experimental Psychology: General*, *113*:121-126.

Compton, K., Kybartas, B. & Mateas, M. (2015). *Tracery: An Author-Focused Generative Text Tool*. In Proc. of International Conference on Interactive Digital Storytelling, Denmark:154-161.

Fauconnier, G. & Turner, M. (2002). *The Way We Think. Conceptual Blending and the Mind's Hidden Complexities.* Basic Books.

Fishelov, D. (1992). Poetic and Non-Poetic Simile: Structure, Semantics, Rhetoric. *Poetics Today*, 14(1), 1-23.

Ghosh, A., Li, G., Veale, T., Rosso, P., Shutova, E., Barnden, J., & Reyes, A. (2015). SemEval-2015 Task 11: Sentiment Analysis of Figurative Language in Twitter. *In the Proc. of SemEval-2015, the 9th International Workshop on Semantic Evaluations,* Denver, Colorado, June 4-5, 2015.

Ghosh, A. & Veale, T. (2016). Fracking Sarcasm with Neural Networks. In Proceedings of *WASSA'2016, 7th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, at NAACL-2016, San Diego, USA

Ghosh, A., & Veale, T. (2017). Magnets for Sarcasm: Making sarcasm detection timely, contextual and very personal. In *Proc. of EMNLP'2017, the Conf. on Empirical Methods in Natural Language Processing*. Copenhagen, Denmark, September 2017.

Giora, R., Givoni, S., & Fein, O. (2015). Defaultness reigns: The case of sarcasm. *Metaphor and Symbol, 30/4*, 290-313.

Grice, H. P. (1978). Further Notes on Logic and Conversation, In P. Cole (ed.), *Syntax and Semantics, Vol. 9: Pragmatics* (pp. 41-57). New York: Academic Press.

Hao, Y. & Veale, T. (2010). An Ironic Fist in a Velvet Glove: Creative Mis-Representation in the Construction of Ironic Similes. *Minds and Machines 20*(4), 635–650.

Kierkegaard, S. (1841). *The concept of irony with continuous reference to Socrates*. Harper & Row, 1966 reprint, p. 278.

Kreuz, R. J., & Glucksberg, S. (1989). How to be sarcastic: The echoic reminder theory of verbal irony. *Journal of Experimental Psychology: General*, *118*(4), 374-386.

Kumon-Nakamura, S., Glucksberg, S., & Brown, M. (1995). How about another piece of pie: the allusional pretense theory of discourse rrony. *J. of Experimental Psychology: General*, *124*, 3-21.

Reyes, A., Rosso, P. & Veale, T. (2013). A multidimensional approach for detecting irony in twitter. *Language Resources and Evaluation 47:239--268.*

Riloff, E., Qadir, A., Surve, P., De Silva, L., Gilbert, N. & Huang, R.. (2013). Sarcasm as contrast between a positive senti- ment and negative situation. In *EMNLP*, vol. 13:704–714

Sperber, D., & Wilson, D. (1981). Irony and the use-mention distinction. In P. Cole (ed.), *Radical Pragmatics* (pp. 295–318). New York, NY: Academic Press.

Taylor, A. (1954). Proverbial Comparisons and Similes from California. *Folklore Studies* 3. Berkeley: Uni. of California Press.

Valitutti, A. & Veale, T. (2015). Inducing an Ironic Effect in Automated Tweets. In Proceedings of the Sixth International Conference on Affective Computing and Intelligent Interaction (ACII2015), September 21-24, Xi'an, China

Veale, T. (2011a). We Can Re-Use It For You Wholesale Serendipity and Objets Trouvés in Linguistic Creativity. In Proc. of the 2nd International Conf. on Computational Creativity, Mexico.

Veale, T. (2011b). Creative Introspection and Knowledge Acquisition: Learning about the world thru introspective questions and exploratory metaphors. In Proc. of AAAI'2011, the 25th Conf. of the Assoc. for Advancement of AI, San Francisco, CA.

Veale, T. (2013). Humorous Similes. *HUMOR: The International Journal of Humor Research, **21**(1):3-22.*

Veale, T. (2015). The Humour of Exceptional Cases: Jokes as Compressed Thought Experiments. *In Brône, G., Feyaerts, K.. and Veale, T. (eds.), Cognitive Linguistics and Humor Research: Applications of Cognitive Linguistics*, Mouton de Gruyter, 69-90.

Yla R Tausczik & J W Pennebaker. (2010). The psychological meaning of words: Liwc and computerized text analysis methods. *Journal of language and social psychology*, 29(1):24–54.

Appointment in Samarra:

Pre-destination and Bi-camerality in Lightweight Story-Telling Systems

Tony Veale
School of Computer Science and Informatics
University College Dublin, Belfield D4, Ireland.
Tony.Veale@UCD.ie

Abstract

Stories are most able to sweep us up and carry us along when we design them to be journeys of the mind. This paper presents a unification of two journey-based story generation models, the character-development model of *The Flux Capacitor* and the plot development model of *Scéalextric*. This union of complementary approaches allows us to build stories with shape *and* directionality. Moreover, since it facilitates the generation of coherent stories by the most minimal of computing architectures, the memory-less state machine, this joint model proves to be ideally suited to the generation of stories by bots. To squeeze a full story-generator into the context-free grammars of Tracery, we give a practical form to two exotic ideas: predestination, and bicamerality of mind.

Journey into Mystery

Every story is a journey we willingly undertake, especially when in the company of relatable characters and an adroit guide. Most are forays into the unknown, as only an author can lead the way to our final destinations. Our stories lay down these paths to other lives by instantiating a metaphor schema Lakoff and Johnson (1980) call *Life is a Journey*, and what Yorke (2013) calls – using another metaphor that shapes many a tale – a journey *into the woods*. Campbell (1949) saw this journey as the monomythic basis of most heroic tales: when heeding the call to adventure, heroes must leave behind the world of the familiar to meet new challenges in strange new lands. Only when they have been changed by their experiences can heroes ever return home, to find themselves and their old lives utterly transformed.

We have good reason for talking of the *twists and turns* of a thrilling tale, for twisty tales arise from journeys along twisted tracks. Authors sometimes propel their characters along paths with unexpected destinations, for reasons that only become clear at the very end of a journey. Consider this tiny gem from the master of the short story, Somerset Maugham (1933). The entirety of the tale is given below:

> "The speaker is Death.
>
> There was a merchant in Baghdad who sent his servant to market to buy provisions and in a little while the servant came back, white and trembling, and said, Master, just now when I was in the marketplace I was jostled by a woman in the crowd and when I turned I saw it was Death that jostled me. She looked at me and made a threatening gesture. Now, lend me your horse, and I will ride away from this city and avoid my fate. I will go to Samarra and there Death will not find me. The merchant lent him his horse, and the servant mounted it, and he dug his spurs in its flanks and as fast as the horse could gallop he went. Then the merchant went down to the marketplace and he saw me standing in the crowd and he came to me and said, Why did you make a threatening gesture to my servant when you saw him this morning? That was not a threatening gesture, I said, it was only a start of surprise. I was astonished to see him in Baghdad, for I had an appointment with him tonight in Samarra."

As Scrooge tells us in *A Christmas Carol*, "Men's courses will foreshadow certain ends ... but if courses be departed from, the ends will change." Tales of predestination, such as Maugham's, subvert this logic with characters who rush headlong toward the inevitable as they run from their fates. In truth, all fictional characters are subject to the forces of predestination; what differs from tale to tale is the extent to which authors reveal the shape of the tracks on which their characters are forced to run, and whether or not characters have any self-knowledge of those tracks. Automated story-tellers are no less *natural* then in their use of rigid plotting and goal-driven planning than their human counterparts. In this paper we argue it makes sound computational sense to explicitly model this notion of character predestination. We will show how predestination can simplify the construction of dense narrative spaces to a point where coherent stories can be generated with the simplest context-free grammars.

Our goals here are more practical than empirical: we aim to simplify the mechanics of story-telling to a level where complex stories can be woven by a minimal state machine with no memory and no global executive. To this end we rehabilitate another somewhat exotic idea, Jayne's (1976) theory linking consciousness to the bicamerality of mind.

For Jayne, the flow of data between the hemispheres of the brain is an interior dialogue that only becomes an internal *monologue* when beings become conscious enough to take full ownership of both sides of the conversation. We do not set out here to tackle the grand challenge of consciousness, for as Jayne notes, it is not at all clear that consciousness is even needed for creativity. Yet Computer Science makes many bicameral divisions that are usefully blurred by AI, such as the line between code and data that is erased by the LISP and PROLOG languages, and we will show here how simple generative systems can weave stories by sustaining a back-and-forth dialogue between simpler bicameral parts.

We unite these strands in the following sections, starting with a discussion of related work and ideas in the next. Our purpose is to unify two complementary approaches to story creation that focus, respectively, on character development and plotting: the *Flux Capacitor* of Veale (2014) and the *Scéalextric* model of Veale (2017). We show here that the unification of both permits the construction of dense story spaces in which characters may wander, not idly or blindly, but with a sense of purpose and narrative momentum. As labeled directed graphs, these spaces are easily transformed into lightweight *Tracery* grammars (Compton *et al*., 2015), which can then be used to specify generative Twitterbots. The advantages of the context-free *Tracery* formalism outweigh its expressive limitations, and we show here how the idea of predestination proves to be a practical workaround to the need for long or short-term memory. To also obviate the need for top-down planning in story telling, we show how Jayne's bicameral divide finds a practical counterpart in the two-grammar approach to bot definition of George Buckenham's *CheapBotsDoneQuick.com*, a web platform that hosts Twitterbots specified as Tracery grammars. So we model story generation as a two-level process in which we first build generators of story *spaces*, and then specify context-free explorers of these spaces to generate novel stories as they race to their own appointments in Samarra.

Related Work and Ideas

The journey schema is so conducive to story-generation by a machine not just because it offers a productive metaphor for narratives; it is also a productive metaphor for AI itself, or at least AI in the classic search-oriented mold. Just as a hero searches for resolution on some Campbellian quest, or roams the narrative thicket of Yorke's woods, AI problem solvers purposefully explore a state-space of possibilities, backtracking here and advancing there, until a predefined objective is reached. Creative systems are free to alter their objective functions – their sense of *value* – as they wander, just as they might transform the space itself. In either case, the need for search persists. For a story-telling AI the space is a graph of branching narrative possibilities, and the story is a function of the path taken by the teller to its goal state. This story-path can be given an *a priori* rationale *post-hoc*, to justify the actions of a hero in terms of their end state, as though the hero planned the actions to reach that very state. Or this rationale can be specified *a priori*, so that a planner can then seek the most dramatic path to making it a reality. Riedl & Young (2010) thus use an explicit planner to give their heroes issues to resolve and the plans to resolve them, yet most story-generation AI systems, from Meehan (1981) and Turner (1994) to Pérez y Pérez & Sharples (2004) to Riedl and Young (2010) to Gervás (2013) and Gervás *et al.* (2016) string together causes and effects to construct plots that seem to imbue characters with plan-like intentionality.

We read intentionality into the way a character interacts with others. If A assists B to reach C then reaching C may have been A's goal all along. The bric a brac of a story are its ancillary figures, obstacles, signs, magic talismans, its helpers and hindrances on the road to its final destination. In exploiting the affordances of these narrative morphemes – what Propp (1928) calls the *morphology of the tale* – a hero exhibits relatable drives and intentions. Propp applied his morphological analysis to Russian folktales, but authors such as Gervás *et al.* (2016) have applied his inventory of character types and functions to the generation of more modern narratives. Others focus on specific elements of the Proppian scheme. Veale (2014) sees the transformational role of stories – how they turn characters of type A into heroes or villains of type B – as the most fascinating aspect of story generation. Propp applied the label *transfiguration* to the transformation of a hero in a story, whilst Campbell dedicated several key stages of his hero's journey to the change, from the call to adventure and the crossing of the threshold to the midway ordeal and near-end resurrection.

Veale (2014) defined a Campbellesque annotation for use in the *Flux Capacitor* to label the actions we typically associate with people from different categories, from artists and scientists to priests and criminals. Every category can be viewed as a journey, with the *call to adventure* serving as its entry point, and the *ordeal* (after a trip to the *inmost cave*) serving as its point of egress. Actions of the first kind are annotated as level 0 when they initiate a person into a category; for instance, studying medicine is a level 0 action for doctors whilst renouncing religion is a level 0 action for atheists. Actions of the second kind are annotated as level 9 if they result in an erstwhile member breaking fully with a category; finding religion is a level 9 action for atheists, whilst losing religion is a level 9 action for believers. The labels 1 to 8 are reserved for actions that link the extremes, with 5 representing the high-water mark of a category, the point at which a person is fully operational as a member; for example, the act of evangelizing as a believer, treating illness as a doctor or spreading doubt as an atheist. Actions labeled with a 2, 3, 4 or 5 mark the growth of a character, while a 6, 7 or 8 document the character's gradual move to the exit. The *Flux Capacitor* generates its plots by linking an exit from one category with an entry into another, and pairs its categories so as to maximize affective dissonance. So, in this way, atheists become believers, heroes become tyrants, sinners become saints, billionaires become bums and cops turn into the crooks they most despise. As such, *Flux Capacitor* generates capsule tales with an ironic shape, mere plot outlines rather than fleshed-out narratives.

The *Scéalextric* model of Veale (2017) focuses more on the bread-and-butter issues of plot design: given an action

V by character A toward character B, with what action is B likely to respond? Given a suitable response V', a system can now determine how A might respond with V'', and so on, until a terminating action V* is performed by A or B. A causal graph of actions and reactions was first constructed by looking for pairs of annotated actions in *Flux Capacitor* with sequential labels, such as 0,1 or 6,7, and by linking these actions into a labeled directed graph. When the first action's label is in {0…5} and the second's is in {6…9} then the connecting arc is labeled "but" in the causal graph; it is labeled "then" in all other cases. This initial graph is manually edited to transform many "then" labels into "so" labels when the connection is a strongly causal one. At this stage additional arcs are also added to create a dense story graph in which 820 different action "verbs" are interlinked. To generate a story, a generator picks a verb at random and initiates a random walk in the forest of causal connections. For every action in the graph, a piece of text is defined to serve as a scene-setter for a story opening with that action. A short text is likewise defined for every action to serve as a moral summation should a story terminate at that action. Also associated with each action is a set of one or more idiomatic templates, to allow each to be rendered in fluent natural language. Any random walk in the causal graph can thus be framed as a complete narrative, with a motivating introduction and a summarizing conclusion bookending a locally-coherent journey along causally-connected actions.

When plotting is reduced to a random walk in the causal woods, characterization fulfills an ever more vital function. Characters may follow a plot as it winds through the graph, but readers will only follow those characters if they seem to know what they are doing. To achieve an integration of character and plot, a system must either choose its actions to suit a character, or it must at least render those actions to reflect what readers already know about the characters. The experiments of Veale & Valitutti (2017) evaluate the latter. Using *Scéalextric* to generate a range of plots, they render the plots as textual narratives using two alternate strategies. In the first, character labels are chosen at random from a pool of stock animals, such as koala, monkey and snake, and plots are rendered by inserting these labels (e.g. "*the koala*") into the slots in *Scéalextric*'s idiomatic templates. In the second, familiar characters are plucked from a large inventory of famous faces, fictional and historical, called the *NOC list* (Veale, 2015). This knowledge-base describes its characters in generous detail, providing for each a list of positive and negative qualities, a set of categories, a list of domains, typical activities, weapons, vehicles and clothing, known opponents and mates, political leanings, and so on. Characters are chosen at random, but in pairs, for each tale, so that the protagonist and antagonist are well-matched and perhaps thematically-related too. Steve Jobs might thus be paired with Leonardo Da Vinci or Bill Gates. When actions involving NOC characters are rendered, the system tries to shoehorn specific knowledge from their NOC entries into the text; for example, if A attacks B, the weapon of choice for A is used; when B flees from A, the vehicle of choice for B is used, as is an associated location to hide in.

Evaluating the outputs of each strategy on 6 dimensions – *laughter*, *entertainment*, *imagination*, *vividness*, *drama* and *silliness* – using the crowd-sourcing site CrowdFlower, Veale & Valitutti reported significant improvements for all dimensions when plots are rendered with NOC characters as opposed to generic animals. Strikingly, this applies just as much to *drama* – the dimension that is, most obviously, the product of plot-level decisions – as it does to any other. In the next section we take the road not followed by Veale & Valitutti, to explore the other approach to the integration of characterization and plot: picking (as opposed to merely rendering) a story's actions to suit the characters involved.

Lost in Narrative Space

The *Flux Capacitor* maps actions to the kinds of characters that perform them, while *Scéalextric* maps actions to each other, to yield a narrative model of cause and effect. Since character influences actions and actions shape character, it makes sense to unify these complementary approaches. To put plot at the service of character, we can use *Scéalextric* to search for the shortest sequence of actions that produces, and explains, any change proposed by the *Flux Capacitor*. Conversely, to use character to drive plot, we can use the *Flux Capacitor* to specify the first and last actions of a plot and use *Scéalextric* to trace out the intermediate journey.

Scéalextric assumes that each of its stories involves just two principal characters, a protagonist A and antagonist B, so its various structures and templates have slots to house the character choices that are ultimately made for A and B. The *Flux Capacitor* makes similar assumptions about arity: categories are associated with actions that comprise a verb and another category, such as *heal:illness* and *debate:idea*. When the other category denotes a kind of person, the verb may denote an interpersonal relationship, such as *criticize* or *debate_with*, that is also defined for *Scéalextric*. In those cases we can map the categories connected by the verb into two roles, the protagonist (A) and antagonist (B). Any verb linking A and B that is annotated with a 0, 1 or 2 can now be used as the opening action of a story involving A and B, while a connecting verb annotated with an 8 or a 9 (not every category has a level 0 verb or a level 9 verb) can be used as the closing action for the same story. Consider the example of *theorist* and *critic*, which are linked by the verbs *disagree_with* (level 1) and *denounce* (level 8). The level 0 action for theorist, *develop:theory*, is not one that can be exploited by *Scéalextric*, so we must settle for one annotated as level 1. Likewise, the denunciation of a critic does not usher a person out of the theorist category, so this action is annotated as level 8 rather than level 9. However, *denounce* is a verb that is also defined for *Scéalextric*, so it makes a suitable destination for any story about a theorist. Using *Scéalextric* to trace out a path from *disagree_with* to *denounce,* the following sequence of actions is proposed:

disagree_with → *are_debated_by* → *are_roused_by* → *fall_in_love_with* → *confess_to* → *are_betrayed_by* → *are_arrested_for_killing* → *denounce*

This shows precisely what *Scéalextric* brings to the union of both systems that *Flux Capacitor* cannot provide alone: its journey through the causal graph pushes the relationship between theorist and critic into the realm of romance, with a dark turn into betrayal and retribution. This is just one of many pathways between disagreement and denunciation in *Scéalextric's* causal graph, and other plots can be derived from the same start and end points. These can be rendered with the roles of A and B filled with "the theorist" and "the critic" respectively, or the NOC can be used to suggest some appropriate names to attach to these categories, such as *Rush Limbaugh* as critic and *Charles Darwin* as theorist.

As presented in Veale (2017), all *Scéalextric* stories start and end at arbitrary points in the causal graph. The paths proposed by *Flux Capacitor* yield more interesting stories because they reflect the journeys taken by people through their chosen categories in life. This category-journey gives each narrative a satisfying shape, and directly instantiates Lakoff & Johnson's *Life is a Journey* schema. Taking its cues from *Flux Capacitor*'s annotations, the joint system generates 12,000 stories that start at a category-entry point and terminate at the brink of category-departure. We could generate far more or far less, but this is an ample sample. We then fold these 12,000 pathways into a single directed graph S that will serve as our story space. Each vertex V in S is an action verb that links to the next actions in a story with arcs labeled *so*, *then* or *but*. Unlike the causal graph used by *Scéalextric*, a subset of vertices are marked as start or end nodes for stories; a well-formed story can start at a vertex designated *start* and conclude at one designated *end*. Since the original 12,000 stories are merged, any single vertex leads directly to any of the subsequent actions from any story that contains it. In this way the graph S gives rise to story possibilities that are not in the original sample.

These possibilities include a potential for the story-teller to get lost in the woods, to wander aimlessly in the graph S until it finds a vertex, any vertex, designated *end*. For the teller to explore S with a sense of purpose, every vertex V must act as a signpost, not just to the very next vertices but to the end of the story too, otherwise the shape imposed on those stories by the *Flux Capacitor* will have been lost. To give vertices a sense of predestination, they must encode not just an action itself, but the final action of the story too. Here is our *theorist:critic* plot again, in this new encoding:

disagree_with/denounce → *are_debated_by/denounce* → *are_roused_by/denounce* →*fall_in_love_with/denounce* → *confess_to/denounce* → *are_betrayed_by/denounce* → *are_arrested_for_killing/denounce* → *denounce/denounce*

When our sample of 12000 stories is folded into S with this encoding, every vertex V/E in S carries with it a sense of narrative momentum. A vertex V/E represents the action V in a tale terminating with the action E, so that V/E can only be connected to other vertices V_1/E, V_2/E, …, V_n/E. Thus, any vertex in a story ending with betrayal can lead only to other vertices from tales of betrayal. So from the very start of a story, the teller knows how the tale will end, even if it does not yet know how that end will ultimately be reached.

Release the Bots

When a story graph S encodes long-distance directionality into every vertex V/E, an explorer of S no longer needs its own sense of direction. The territory becomes its own map *and* compass, so an explorer need keep no record of where it has been or where it is going. We can thus turn this map into a formal device that lacks all memory, such as a finite-state-machine. Since the graph S already resembles such a machine, with certain states/vertices marked as permissible start states and others marked as allowable end states, we can translate S directly into the corresponding Chomskyan grammar. Our choice of formalism is *Tracery* (Compton *et al.*, 2015), a JSON-based format for context-free grammars that is widely-used for procedural content generation. The resulting Tracery grammar can be directly given to CBDQ (*CheapBotsDoneQuick*) to create a story-telling Twitterbot.

A Tracery grammar is a set of rewrite rules in which a non-terminal on the left-hand side is replaced by a random choice of expansions from the right-hand-side, as in:

"color": ["red", "blue", "green", "orange", "black"],

An expansion on the right may recursively mention a non-terminal (in hashes) that is then further expanded, as in:

"toy": ["#color# ball", "#color# bike", "#color# doll"],

The following Tracery rule is used by a Trump parody bot, *@trumpScuttleBot*, to tweet satirical *roses are red* poems:

"poem": ["#red_thing# are red, #blue_thing# are blue,
my #fan# #affirmation#, and #blue_rhyme#"],

When other non-terminals such as *red_thing* are defined, our grammar tweets (via CBDQ) the following short poem:

Plastic roses are red,
Sailors' curses are blue,
my human children will build my wall,
and pray that profits ensue

To generate a Tracery grammar from a story graph S, each vertex V/E is defined as a non-terminal with one expansion string for each adjacent next vertex in S. Each expansion string contains an idiomatic rendering (via *Scéalextric*) for its action verb, followed by a non-terminal reference to the set of possible next vertices on the path to a valid endpoint. The exception to this norm is the expansion string for any vertex of the form E/E, such as *denounce/denounce*: since this form indicates the last action in a story, the expansion contains the text "The End" in place of a non-terminal. The set of all vertices in S that can launch a story are gathered together as expansions for a single rule called "origin", the label Tracery reserves for the master rule of any grammar.

Since Tracery rules have no memory of prior expansions they cannot carry forward any context – such as names for the characters A and B – from one rule to the next. As a workaround, we can encode in the expansion of each V/E vertex the pair of categories that inspired a path through that vertex in S. In the following tweet these categories have been wrapped in quotes, and alternate across actions:

A 'master' was resented by a 'rival' and our 'master' overshadowed this 'rival' so our 'trailblazer' was copied by this 'imitator' but our 'tempter' misled this 'sinner' so our 'abbess' was dismissed by this 'bishop so our 'victimizer' begged forgiveness from this 'victim'

The quotes identify the categories as likely metaphors, yet no matter how relevant they may seem for specific actions, most metaphors are ambiguous and readers are easily disoriented as to who is who in this story. Is the *victim* that ends the tale the *master* that begins it, or is this *victim* the *rival*? To avoid confusion and foster narrative momentum, each action should be rendered with the same pair of characters. Yet since each is rendered independently of all others – this is what it means for a grammar to be context-free – we must rely on the sense of direction that is baked-in to each state and non-terminal. Predestination provides the answer: we associate a unique pair of characters (A & B) with each action E that can terminate a story with a vertex E/E. In our sample of 12,000 stories there are 220 distinct verbs that fit the bill, allowing 220 character pairs to be used by the grammar. Given the large inventory of name pairs that is harvested from the NOC list – we collect the first names of characters and their enemies or mates, such as *Woody* & *Mia* and *Sam* & *Diane* – we randomly assign these to the 220 termination actions. Suppose *Woody* & *Mia* is mapped to *beg_forgiveness_from*; all stories that end with this verb, and every action within those stories, will be rendered with A=*Woody* and B=*Mia*. In effect then, Woody is always destined to beg Mia for forgiveness, no matter how a story about them may begin.

While the number of terminating verbs is large, a reader may soon recognize the inevitability of tales with specific characters ending in foretold ways, so that e.g. Sam always marries Diane or Hillary always kills Bill. However, it is a simple matter to regularly regenerate S (once a week, say) and to randomly reassign characters to terminating verbs. Like the actors in a travelling repertory company, who may switch roles from one town or one production to another, the characters in our tales trade destinies with each other. When the new grammar that results from a new S is given to CBDQ, the bot's tales are given a new lease of life too.

Bicameral Bots

Our story above about a master and a rival barely squeaks under Twitter's newly enlarged 280-character tweet limit. To give a story room to breathe, a bot should ideally parcel it into an array of small episodes – say, one action apiece– and emit it as a threaded sequence of individual tweets, the way humans tend to use Twitter for fiction. But this kind of dismemberment would require planning and a global view of the story, and if a Tracery grammar lacks the memory to pass context between non-terminals, it certainly lacks the ability to pass control from one tweet to the next. However, CBDQ makes an interesting bicameral distinction in its use of grammars that offers bot-builders a nonobvious solution.

Twitter is more than a broadcast medium for the sharing of opinionated content; it is also a platform for interaction in which people relate to each other by replying to, and by commenting upon, each other's tweets. Our bots, likewise, are more than deaf generators. We often build these bots to respond to the provocations of others as much as to deliver automated provocations of their own. Buckenham's CBDQ thus allows *two* grammars to be specified for a Twitterbot: a core Tracery grammar, which, as we have seen, generates a bot's outputs on an agreed schedule, free of all influence from the outside world; and a simpler response grammar that allows a bot to reply directly to any mentions of its @ handle in the tweets of others. While also expressed in a JSON format, this second grammar is a not a fully-fledged piece of Tracery. Rather, it amounts to an ordered list of *stimulus:response* pairs: the stimulus is a literal string that any @ mention must contain before the response – a single Tracery expansion string, which may refer to non-terminals in the core Tracery grammar – is used to generate a reply. Suppose, for instance, that we want our Trump parody bot to produce poetry on demand. In response to a color from a user, the bot generates a poem around that color. Consider:

"[g|G]old": "#red_thing# are red, #gold_thing# are gold,
my #fan# #affirmation#, and #gold_rhyme#",

This response rule ensures that tweets to *@trumpScuttleBot* containing 'gold' or 'Gold' receive a response such as this:

Self-inflicted wounds are red,
Goldfinger's ladies are gold,
my local milk people are TREMENDOUS,
and are my special interests (I'm Sold!)

The bicameral parts of a bot can talk to each other in ways that are both simple and roundabout. An incoming tweet is matched to a stimulus in the response grammar, which then *talks* to the core grammar by invoking its non-terminals in the construction of its response. But the core grammar can only *talk* to the response grammar if it addresses its tweets to itself, by appending a mention of its own Twitter handle. Such mentions bring the outputs of the core grammar to the attention of the response grammar, which can then respond in kind, perhaps also appending a self-reference to ensure that the conversation between bicameral halves continues. While this conversation is carried on between its grammars the bot is basically, but quite productively, talking to itself.

The job of an automated story-teller is to spin a tale by talking to itself. To exploit CBDQ's basic bicamerality, we partition and reshape the story-telling grammar as follows. The core grammar again has a non-terminal/rule for every state V/E or E/E in the story graph S, but each right-hand-side expansion no longer contains recursive non-terminals. Instead, each expansion ends with the bot's Twitter handle. The main rule, *origin*, is responsible for generating just the first tweet of the story, a single action with a title, as in:

The 'Nanny' & The 'Child'
The story of how Lois supervised Kal's every effort
@BestOfBotWorlds

The trailing self-reference is later picked up by the bot's response grammar, as CBDQ is attuned to mentions of its bots (even by themselves) on Twitter. The tweet identifies the action verb *supervise* but does not explicitly identify the state in S, *supervise/are_arrested_for_killing*, to which it fully corresponds here. However, recall that the choice of characters Lois and Kal is a function of the very last action, so from their presence in this tweet the response grammar can recover this latent state in S. Here is the response rule:

"Lois supervised Kal": "#a# #*supervise/arrest_f_kill*#",

The non-terminal *a* is a shorthand that expands to the bot's own Twitter handle, which will prepend any bot response. The non-terminal *supervise/arrest_f_kill* is also defined in the core Tracery grammar, with a rule that ties together all the narrative consequences for the corresponding state in S. This would be laborious work if the grammars were hand-generated, as is the case for most Tracery/CBDQ bots, but these grammars are machine-generated. The tale continues and ends with the following self-addressed tweets:

@BestOfBotWorlds But Lois knowingly told lies for Kal

@BestOfBotWorlds Then Kal threatened to expose Lois's darkest secrets

@BestOfBotWorlds But Lois made a heartful appeal to Kal

@BestOfBotWorlds But Kal's insults struck Lois like poisoned darts

@BestOfBotWorlds And the police arrested Lois for her brutal attack on Kal

The End.

A trailing *The End* is provided by the rule for the end-state *are_arrested_for_killing/are_arrested_for_killing*, an end to which the response grammar knowingly does not reply.

The Blackboard Jungle

The generative grammar of a Tracery/CBDQ bot can only talk to itself by using Twitter as an intermediary, for only by posting tweets into its own public timeline can it pass those messages to the response part of its bot persona. This bicameral conversation uses Twitter as a *blackboard* onto which the bot reads and writes its data (Hayes-Roth, 1985; Veale & Cunningham, 1991). But Twitter is a very public blackboard, as well-suited to cooperation between relative strangers as it is between different parts of the same bot. A bot may thus delegate tasks or provide inspiration to others by sharing and appropriately addressing its ideas in public.

Consider a CBDQ bot, named *@MovieDreamBot*, which scours the category hierarchy of *DBpedia.org* to find ideas for its tweets. The bot targets fictional categories of films and books, exploiting the linguistic form of each to extract the key ideas underpinning a specific work. For example, *Blade Runner* (1982) is listed with the following categories at dbpedia.org/page/Blade_Runner: *flying_cars_in_fiction, climate_change_in_fiction*, *films_about_altered_memories* and *genetic_engineering_in_fiction*. If we now strip away the syntactic sugar, we are left with the themes *altered memories*, *genetic engineering*, *climate change* and *flying cars*. The bot chooses two themes to package into every tweet, of which the following is a representative example:

Influenced by the film 'Blade Runner,' I dreamt of amnesiacs who lose altered memories and drive flying cars, @MovieDreamBot.

The bot combines fictional themes from *DBpedia.org* with propositional content from *Flux Capacitor*, to reason that even flying cars need chauffeurs, just as the memories that are lost by amnesiacs may have been altered in some way. Note how the bot addresses itself with an @ self-reference; this allows its response grammar to engage with each tweet and add some further creative value in the process. In this case the response grammar is designed to use a theme from the tweet as the basis for an automated story. Keying off of the term *amnesiacs*, the response grammar responds with:

Hey @BestOfBotWorlds, spin us a yarn about how our amnesiac remembered this particular friend.

So the response grammar does not pass the ball back to its generative partner, but passes it onward to another bot, our story-telling CBDQ bot *@BestOfBotWorlds*. As it does so it shifts the emphasis from amnesiac to the *Scéalextric* verb *remember*, allowing the story-teller to reply with this tale:

The 'Freedom Fighter' & The 'Martyr'
The story of how Rick held on to memories of Ilsa
@BestOfBotWorlds

@BestOfBotWorlds And Ilsa filled Rick with inspiration

@BestOfBotWorlds So Rick heard wedding bells when looking at Ilsa

@BestOfBotWorlds But Rick made Ilsa sick to her stomach

@BestOfBotWorlds So Ilsa kicked Rick out into the cold

@BestOfBotWorlds So Rick whispered rumours behind Ilsa's back

@BestOfBotWorlds Then Rick hurled cruel taunts at Ilsa

@BestOfBotWorlds Then Ilsa rose up against Rick

@BestOfBotWorlds Then Ilsa toppled Rick from the top of the heap

@BestOfBotWorlds Yet Rick became a shining inspiration for Ilsa

@BestOfBotWorlds But Rick crucially underestimated Ilsa

@BestOfBotWorlds And Ilsa knew just how to manipulate Rick

@BestOfBotWorlds So Rick caught a bullet to save Ilsa
The End.

We humans throw ideas about on social media as though they were balls to be volleyed with great force and sliced with spin, and our bots should be able to do the same. For a story emerges from several distinct layers of interaction: the interplay of words and ideas, the interplay of teller and audience, and the interplay of fictional characters. Our bots and their grammars can interject themselves into each kind of interaction, to collectively create the Twitter equivalent of what Minsky (1986) called *the society of mind*. For even if each mindlessly executes a tiny task of its own, our bots can cumulatively give rise to surprisingly creative results.

West of Eden

Predestination is a recurring fictional trope that is found in movies, novels, TV shows and games. By comparison, the bicamerality of mind has remained the stuff of esoterica, at least until now. *Westworld*, a recent HBO television series, has put both ideas side-by-side in the popular imagination. Like the 1973 movie of the same name on which the series is based, *Westworld* is set in a Western-styled theme park where lifelike robotic "hosts" – in the guise of cowboys, barmaids, lawmen, thieves, farmers and cathouse madams – entertain paying guests with their highly scripted antics. *Westworld* is a technological marvel, overseen by operators who are as much story-tellers as roboticists or bureaucrats. Yet no matter how lifelike and conscious a host may seem, each is predestined to traverse the same narrative "loop" to reach its appointed fate. While each is given some latitude for improvisation within its loop, every host is always fated to be on time for its appointment in Samarra. The hosts are dancers that collectively glide through a highly-structured story-space, one that is regularly regenerated and rebooted with new loops, new fates, and new roles for old hosts.

In episode 3, season 1, the park's chief designers discuss the bicameral basis of the host's mental architectures. The younger designer sums up, and dismisses, Jayne's theory thusly: "the idea that primitive man believed his thoughts to be the voice of the gods, but I thought it was debunked," to which the older replies "as the theory for understanding the human mind perhaps, but *not* as a blueprint for building an artificial one." The distinction, so well articulated in this work of modern fiction, is one that has always been present in the field of Computational Creativity. Cognitive theories offer valuable insights into the working of creative systems but they need not always hold water for human cognition to be of practical value to the builders of artificial systems. In many ways the practitioners of CC are guided as much by stories as by theories. While theories come and go, a good story always retains its ability to inspire and to guide.

A subtle philosophical thread that is woven through the *Westworld* series is the possibility that the park's human operators are no more the possessors of a conscious mind and a soul than their robotic creations. Some hosts seem to be more human that their creators, while some guests, and some creators too, are stuck in loops of their own making. So to what extent is any CC system stuck in a loop of its designer's making, and what capacity does such a system have to transcend its programming? We all travel in ruts of society's making, improvising locally within loops that we cannot always see. Shouldn't our CC systems do likewise?

Our engagement with such themes in this paper has been largely superficial, focusing as we have on practical issues in the lightweight design of distributed CC systems. Yet our treatment of practical issues is still usefully informed by a consideration of more profound questions. Our model of automated story-generation aims to reconcile the loops with the improvisations to yield predestination with choice. As builders of these CC systems we become *meta*-tellers; for we build the story-spaces in which our hosts wander on unseen tracks, telling stories as they move around in loops. When a space has exhausted its potential to surprise, we regenerate it, with new paths and new character destinies.

Authorship in a Bottle

We have presented a number of new resources in this paper that are available for download by researchers. Bicameral grammars for our story-telling bots *@MovieDreamBot* and *@BestOfBotWorlds*, as well as for *@TrumpScuttleBot*, can be accessed via links on their Twitter pages or downloaded from the CC repository *github.com/prosecconetwork*. Code and data for the generation of these grammars, as well as knowledge representations for *Scéalextric* and the NOC list are also available from this repository. A forthcoming book on CC Twitterbots (Veale & Cook, 2018) offers greater detail on how these resources may be used by bot-builders.

Squeezing a fully functional story-teller into the expressive confines of a finite-state-machine, or even into the context-free grammars of *Tracery*, is the equivalent of squeezing a seaworthy ship into a bottle. Our focus in this paper has not been on improving the quality of the stories generated by *Scéalextric* or *Flux Capacitor*, even if we have shown how these two approaches can be unified to imbue stories with a greater sense of shape and completeness. Rather, we have focused on replicating the coherence and richness of stories from (Veale, 2017; Veale & Valitutti, 2017) with a more streamlined representation and a much reduced algorithmic complexity. We have shown how bots can use Twitter as a blackboard to distribute creative effort, and how we can squeeze the most from tools such as Tracery & CBDQ by automating the construction of grammars. The philosopher Daniel Dennett (2007:95) once remarked that "we have a soul, but its made of lots of tiny robots." We have set out to model nothing so grand as the soul here, except perhaps the soul of a new story-telling machine, made of tiny bots.

References

Campbell, J. (1949). *The Hero with a Thousand Faces.* Princeton: Princeton University Press.

Compton, K., Kybartas, B. & Mateas, M. (2015). *Tracery: An*

Author-Focused Generative Text Tool. In Proc. of International Conference on Interactive Digital Storytelling, Denmark:154-161.

Cunningham, P. & Veale, T. (1991). Organizational issues arising from the integration of the lexicon and concept network in a text understanding system. In Proc. of the 12th Int. Joint Conference on Artificial intelligence (IJCAI), Sydney, Australia, 981-986.

Dennett, D.. (2007). My body as a mind of its own. In D. Ross (Ed.), *Distributed cognition and the will: individual volition and social context*. Cambridge, MA: MIT Press.

Gervás, P. (2013). Propp's Morphology of the Folk Tale as a Grammar for Generation. In *Proceedings of the 2013 Workshop on Computational Models of Narrative*, Dagstuhl, Germany.

Gervás, P., Hervás, R., León, C. & Gale, C.V. (2016). Annotating Musical Theatre Plots on Narrative Structure and Emotional Content. In *Proc.* of the *7th International Workshop on Computational Models of Narrative*, Krakow, Poland.

Hayes-Roth. B. (1985). A blackboard architecture for control. *Artificial Intelligence*. **26** (3): 251–321.

Jaynes, J. (1976). The origins of consciousness in the breakdown of bicameral mind. Middlesex, UK: Penguin books.

McKee, R. (2010). *Story: Style, Structure, Substance, and the Principles of Screenwriting*. New York: Harper-Collins.

Meehan, J. (1981). *TALE-SPIN*. In Shank, R. C. and Riesbeck, C. K., (eds.), *Inside Computer Understanding: Five Programs plus Miniatures*. Hillsdale, NJ: Lawrence Erlbaum.

Minsky, M. (1986). *The Society of Mind*. Cambridge MA: MIT Press.

Pérez y Pérez, R. & Sharples, M (2004). Three Computer-Based Models of Storytelling: BRUTUS, MINSTREL and MEXICA. *Knowledge Based Systems Journal,* 17(1):15-29.

Propp, V. (1928/1968). *Morphology of the Folktale*. University of Texas Press (2nd edition; English translation by Laurence Scott).

Riedl, M. O. and Young, R. M. (2010). Narrative planning: balancing plot and character. *Journal of Artificial Intelligence Research* 39.1, 217-268.

Schank, R. & Abelson, R. (1977). *Scripts, Plans, Goals and Understanding*. Psychology Press: New York, NY.

Turner, S.R. (1994). *The Creative Process: A Computer Model of Storytelling*, Hillsdale, NJ: Lawrence Erlbaum.

Veale, T. (2014). Coming Good and Breaking Bad: Generating Transformative Character Arcs For Use in Compelling Stories. In *Proceedings of ICCC-2014, the 5th International Conference on Computational Creativity, Ljubljana, Slovenia, June 2014.*

Veale, T. (2016a). Round Up The Usual Suspects: Knowledge-Based Metaphor Generation. In Proceedings of *the Meta4NLP Workshop on Metaphor at NAACL-2016, the annual meeting of the North American Assoc. for Comp. Ling. San Diego, CA*.

Veale, T. (2016b). A Rap on the Knuckles and a Twist in the Tale: From Tweeting Affective Metaphors to Generating Stories with a Moral. In *Proceedings of the AAAI Spring Symposium on Ethical and Moral Considerations in Non-Human Agents.*

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 Computational Creativity, Atlanta, Georgia, June 19-23*.

Veale, T. & Valitutti, A. (2017). Tweet dreams are made of this: Appropriate incongruity in the dreamwork of language. *LINGUA 197, 141--153.*

Veale, T. & Cook, M. (2018). *Twitterbots: Making Machines that Make Meaning*. Cambridge, MA: MIT Press (in press).

Vogler, S. (1984/1998). *The Writer's Journey: Mythic Structure For Writers*. Studio City, CA: Michael Wiese Productions (a book treatment of Vogler's original 7-page memo from 1984).

Yorke J. (2013). *Into the Woods: A Five-Act Journey into Story*. London, UK: Penguin.

Towards Goal-aware Collaboration in Artistic Agent Societies

Otto Hantula and Simo Linkola
Department of Computer Science
University of Helsinki
{otto.hantula, simo.linkola}@helsinki.fi

Abstract

We study the effects of goal-awareness in artistic agent societies creating evolutionary art. Particularly, we examine how goal-awareness may be utilized in modeling an agent's peers when the aesthetic goals of the agent and its peers are subject to change. The agents use the learned peer models to choose their collaboration partners, and may alter their own aesthetic goal for the duration of the collaboration in order to enhance the potential of the collaboration outcomes. In addition, we demonstrate how goal-awareness can be used to guide the aesthetic goal change. The empirical evaluation indicates that agents which can adapt to their collaboration partners are more likely to reach favorable collaboration outcomes, even when their partners perceive fundamentally different properties from the artifacts.

Introduction

An agent seeking to select suitable collaboration partners in a creative society where the agent's and its peers' *aesthetic goals* are subject to change raises the need for dynamic peer models. We study how *goal-awareness* (Linkola et al. 2017), the ability to monitor and control one's own goals, can be utilized in peer modeling and collaboration partner selection, and to facilitate favorable collaboration outcomes. Further, we demonstrate how an agent can use goal-awareness in conjunction with novelty-seeking (curious) behavior to strategically change its own aesthetic goals. In empirical evaluation, we observe that goal-aware agents are more likely to reach favorable collaboration outcomes and strategic aesthetic goal change causes emergent phenomena encompassing the whole society.

We build upon our earlier work (Linkola and Hantula 2018), where we investigated how artistic agents creating evolutionary art (Sims 1991; Romero and Machado 2007) could find feasible collaboration partners in a society consisting of agents with different skills and aesthetic preferences. Each agent aims to produce both valuable and novel outputs (Boden 1992) using aesthetic measure for value and a memory of previously seen artifacts for novelty. The agents interact with each other through artifact exchanges and pairwise collaboration, in which the agents aim to jointly create an artifact. To distinguish favorable collaboration partners, each agent learns a private model of its peers, which it then utilizes in collaboration partner selection.

In our previous work the agents had static aesthetic preferences which they couldn't change. In this paper, we expand the problem setting by allowing each agent's aesthetic goals to change, resulting in a dynamic and more complex situation. To handle the increased complexity, we provision the agents with different traits of goal-awareness.

Drawing from self-adaptive (see, e.g. Salehie and Tahvildari (2009)) and self-aware systems (Lewis et al. 2015), our main focus is in metacreativity (Linkola et al. 2017). First, we are interested in how goal-awareness may benefit an agent in selecting its collaboration partners in dynamic situations where its own and its peers' preferences are subject to change. Second, the goal-aware agents have an ability to adapt their aesthetic goals to a given collaboration partner for the duration of the collaboration. Third, we demonstrate how an agent can use its memory and goal-aware peer models in order to make strategic changes to its aesthetic goal satisfying both its collaboration- and novelty-seeking goals.

The rest of the paper is organized as follows. In the next section, we give motivation and background for our work. Then, we describe our agent society in general and different components of individual agents. We then move to our contributions. First, we go through on a conceptual level the distinct ways we utilize goal-awareness in this paper. Next, we define a new goal-aware peer modeling scheme which exploits the linear nature of our selected aesthetic functions, and describe how an agent can make the strategic aesthetic goal changes using the goal-aware peer model. Then, we outline our empirical experiment setup and present the main results from the experiments. We finish with discussion and conclusions.

Background

Our paper studies social behavior of artistic agents, and we are interested in emergent phenomena during the agent society's lifetime, in the context of computational social creativity (see, e.g. Saunders and Bown (2015)). A prominent conceptualization of social creativity is the system's view of creativity (Csikszentmihalyi 1988). It describes how the accumulated cultural artifacts, i.e. *the domain*, the experts of a given *field* and (each) *individual* are in constant interaction and affect each other. The major claim of the system's view is that creativity is not in any single component but in the interaction between all three components. In this paper, we focus on the individuals' ability to find suitable collaboration partners within the changing field, and how strategic behavior of individuals may cause emergent macro-level phenomena in the society. The domain is modeled implicitly as the collection of artifacts in the agents' memories.

Agent-based simulations have been extensively utilized to study social aspects of creative phenomena. For example, Saunders and Gero (2001) report emergence of communication cliques between agents with matching hedonic functions in a society of curious agents producing evolutionary art; Sosa and Gero (2005) reveal emerging social roles, such as gatekeepers, and other social phenomena when simulating designers and their societies; and Gabora and Tseng (2014) show that self-regulation of new ideas may have a positive effect on the mean fitness of ideas present in an agent society.

However, in agent-based simulations, interaction between the agents is typically defined using simple rules. The agents may be (directly) affected by the choices and actions of their neighbors or the society as a whole, but they do not model distinct peers in order to make strategic decisions about their own behavior involving those peers.

On the other hand, the skills, preferences and other properties of the collaboration participants have a direct result on the collaboration outcomes (Uzzi and Spiro 2005). To be able to distinguish favorable collaboration partners or otherwise act with social intent, an agent has to have a model of its peers' "minds" (Castelfranchi 1998).

Collaboration is essential for (computational) creativity allowing the participants to produce artifacts they could not by themselves (Paulus and Nijstad 2003; Uzzi and Spiro 2005; Pérez y Pérez et al. 2010). In computational creativity, collaboration of independent creative agents has gathered the most attention in musical domain. However, even in the musical domain, the set of collaborating agents is typically fixed, e.g. to ensembles where each agent plays a different instrument (Eigenfeldt et al. 2017).

Overall, there is a prominent lack of research considering how independent creative agents should model their peers' collaboration potential and utilize the peer models in their decision making, e.g. when selecting collaboration partners.

Peer modeling becomes a dynamic problem if an agent or its peers are subject to change as time elapses. However, this is the standard situation in creative societies: agents evolve in their style, aesthetic preferences and other properties.

In computational creativity, being aware of one's own creative process and being able to adjust it is often called metacreativity (Linkola et al. 2017). A particularly eminent aspect of metacreativity is goal-awareness. In conjunction with interaction-awareness, a goal-aware agent is provided with tools to adapt to its collaboration partners and change how it perceives its peers in a significant manner. Particularly, an agent may envision how it would observe artifacts if its goals would be different. The agent may then utilize this knowledge and temporarily adapt its goals to a new collaboration partner.

In this work, we hope to take the first steps to address the different concerns mentioned above. Building upon our previous work with interaction-aware agents, we study social behavior of creative agents which interact with their peers intentionally. We aim to add to the understanding of how goal-awareness may aid the agents in their peer modeling and collaboration partner selection, during the collaboration process and in strategically changing their aesthetic goals with respect to how they see their peers.

Agent Society

The agent society consists of a diverse set of artistic agents creating images that are novel and valuable to them. The agents differ in their image creation skills, aesthetic goals and what they are able to perceive in an image. In particular, we are focusing on the effects of changing aesthetic goals through goal-awareness.

The agent society is simulated iteratively. At odd time steps each agent creates a new image individually (we call these *solitary artifacts*). At even time steps the agents pair up and collaborate with their partner, aiming to produce a jointly created artifact. An individual agent can create and evaluate artifacts, as well as interact with its peers by sending them artifacts and through *collaboration*. To guide its interaction with other agents, and possibly other behavior, an agent learns a *peer model*.

An agent creates evolutionary art (Sims 1991) using an *evolutionary engine*. The artifact *evaluation* utilized in the evolutionary engine is based on perceived *value* and *novelty*. An agent has one *aesthetic measure* it uses to compute value and a limited *memory* of seen artifacts it uses to compute novelty. As its aesthetic goal, an agent has a target value for the aesthetic measure. The ability to adjust this target value is the key feature introduced in this paper. When an agent changes its current target value, we call it *movement*.

Next, we move on to describe these abilities and components on a general level. For the full details of the evolutionary engine's configuration and the collaboration process, we refer the reader to Linkola and Hantula (2018).

Evolutionary engine An agent creates a new image using an evolutionary engine, initializing the engine's population partly using the images it has previously made during the simulation. The evolutionary engine uses genetic programming to evolve an expression tree, which is used to calculate the value for each (x, y) coordinate in an image. The tree consists of terminals (leafs) and functions (inner nodes). An agent's image creation skills are determined by the subset of functions it has for creating expression trees.

Aesthetic measure and value For the purposes of this paper we use two aesthetic measures present in our earlier work: entropy and fractal dimension (Linkola and Hantula 2018). Entropy is defined by the color distribution in an image and fractal dimension measures an image's structural properties. Each agent has only one of these two measures, but the actual target within the aesthetic measure's bounds is different for each agent. For the complete descriptions of how the objective values of the aesthetic measures are computed, we guide the reader to den Heijer and Eiben (2014).

The value of an artifact I is calculated based on the evaluating agent's aesthetic measure v and target value, i.e. *aes-*

Figure 1: Example of collaboration between two agents. On the left is a solitary image by a fractal dimension agent, on the right by an entropy agent. On the center is an artifact the agents made in collaboration, showing traits from both agents.

thetic goal g. The closer the objective aesthetic measure calculated from the artifact is to the goal, the more valuable the artifact is. The value is a linear mapping of the distance from the aesthetic goal, calculated with the following formula:

$$\text{value}(I) = \begin{cases} 1 - \frac{|g-v|}{v_{max}-v_{min}}, & \text{if } |g-v| < v_{max} - v_{min} \\ 0, & \text{otherwise,} \end{cases}$$

where v_{min} and v_{max} are the minimum and the maximum values for the aesthetic measurement, respectively.

We chose entropy and fractal dimension as aesthetic measures because of their potential to complement each other. Further, their asymmetrical relationship provides an interesting case for analysis: agents with entropy tend to create images of high complexity regardless of their exact target value, but complexity's target value does not have a strong relation to entropy in the images produced by our agents.

Memory and novelty An agent has memory for up to 500 artifacts, where the agent can store artifacts it has seen. The artifacts can be created by itself or other agents. If the memory is full when storing a new artifact, the oldest artifact in the memory is forgotten.

The novelty of an artifact is evaluated with the function $\text{novelty}(I) = \min_m \text{ed}(I, m)$, where I is the artifact being evaluated, m is an artifact in the agent's memory, and $\text{ed}(\cdot)$ is the normalized Euclidean distance between the artifacts. In other words novelty is the euclidian distance to the closest artifact in the agent's memory.

Evaluation Using the value and novelty calculated from an artifact, an agent uses the following function to get the final evaluation: $\text{eval}(I) = \frac{1}{2}\text{value}(I) + \frac{1}{2}\text{novelty}(I)$.

Movement An agent can change its aesthetic target value, or aesthetic goal, which is used for creating artifacts and selecting collaboration partners. The movement changes what kind of artifacts an agent creates (what they see valuable) and with whom it collaborates. We run tests with two different types of movement. First is completely random, where the new goal is drawn from a uniform distribution. Second utilizes goal-awareness and curiosity in determining the new goal. These are explained in detail later.

Peer model An agent learns a peer model of the other agents from the artifacts they create. The peer model is used to select collaboration partners, to change one's aesthetic goal for collaboration and for goal-aware movement. We use two Q-learning based learning schemes for the peer models, which are described in their own section.

Collaboration In collaboration, a pair of agents merge their artifact creation skills and aesthetic goals aiming to produce an artifact jointly. The collaboration follows an alternating co-creation process (Kantosalo and Toivonen 2016), where the collaboration partners evolve the same artifact set in turns iteratively (see Linkola and Hantula (2018) for details of the collaboration process). Figure 1 shows an example of collaboration between two agents.

If the agents agree on an artifact to be produced as a collaboration's result, we call the collaboration *successful*. If they can't agree on an artifact, no artifact is produced. To negotiate about the collaboration artifact, both agents keep a hall-of-fame of the best artifacts seen during the collaboration process (sorted to an increasing rank, the best artifact having the first rank). At the end of the collaboration, agents compare their hall-of-fames and pick an artifact which has the smallest combined rank as the collaboration result, i.e. they agree on it, if there exists an artifact which is in both hall-of-fames.

Manifestations of Goal-awareness

A goal-aware agent is able to observe how it reaches its current goals and adjust these goals if it sees fit (Linkola et al. 2017). In essence, goal-awareness facilitates creative autonomy (Jennings 2010) in an agent, aiding the agent to change its creative process and produce potentially previously unreachable artifacts.

There are three ways in which our agents can utilize goal-awareness. First, if the agent is aware of the goals of its peers, it can model those goals and their changes and use that information to select feasible collaboration partners with respect to its own current goals. Second, if the agent models its peers' goals, it may adjust its own aesthetic goal for the duration of the collaboration, possibly enhancing the collaboration potential. Third, the agent can use the learned peer models to make strategic changes to its aesthetic goal.

Next, we describe on a conceptual level the different ways in which goal-awareness is implemented in this work.

Peer modeling Changing aesthetic preferences of the agent and its peers imposes new challenges on peer modeling. The peer model has to contain sufficiently accurate and topical information about peers for it to have any value as an asset in an agent's decision making.

Goal-awareness provides capabilities to handle changing aesthetic preferences. If an agent is able to imagine how it would perceive a certain artifact if its aesthetic goal would be different, it can keep an alternative peer model for each of its goals. The agent can adjust each of these peer models when it perceives an artifact from another agent. Then, when the agent changes its own aesthetic goal, it can assimilate the alternative peer model most suitable for the current goal without the need to build the peer model from scratch.

Exploiting the alternative peer models is central for the peer modeling scheme proposed in this paper, ga-Q, which is described in detail in the next section.

Adaptation during collaboration Our agents utilize goal-awareness in collaboration by changing their own aesthetic goal to align with the partner's goal. When the collaboration begins, the agent in the collaboration pair that got to select its partner chooses a *temporary goal*, which it uses during the collaboration. The selected partner doesn't change its goal. Selecting the partner perceived as best and then selecting the temporary goal to suit the partner can be seen as a combination of the selfish and altruistic approaches (Linkola and Hantula 2018). First the agent selfishly selects the partner it personally likes most. Then it altruistically adapts its own goal to be the best possible collaboration partner for the other agent.

Strategic movement Our agents use curiosity to guide when and where to move. There are three factors that affect when an agent decides to move: how long the agent has had its current aesthetic goal, how good artifacts it is producing with respect to its current goal and what the other agents are creating. An agent doesn't want to stay in the same place for too long. It wants to produce valuable artifacts, moving if it fails to do so. A place currently being explored by the society is also seen as less interesting.

When the agents move, they use their memory to guide their movement to less explored areas, exhibiting curiosity. The agents utilize goal-awareness by considering their peers' aesthetic goals, trying to move to areas currently unoccupied by other agents, but still having sufficient collaboration potential. The strategic movement is described in detail in its own section.

Peer Modeling

Peer modeling is the basis for intentional interaction between the agents in our experiments. The learned peer model is used to select good collaboration partners, change the agent's aesthetic goal for collaboration and guide movement. Because the agents have dynamic aesthetic goals, the model has to be able to quickly adapt to the changes in the learning agent itself and the peers. To enable some of the aspects of goal-awareness, we describe a peer model that models all of the agent's possible aesthetic goals simultaneously, even though the agent can only have one of them at a time.

The learning scheme for peer modeling we propose here is an extension to the learning scheme called *hedonic-Q* in Linkola and Hantula (2018). Hedonic-Q is based on Q-learning (Watkins and Dayan 1992), which is a common reinforcement learning method, that maintains Q-values for state-action pairs based on received reward. The Q-value for a state-action pair is the expected utility of choosing the action while in the state. Hedonic-Q uses a simplified, stateless version of Q-learning, with update rule $Q(a_i) \leftarrow Q(a_i) + \lambda(r - Q(a_i))$ (Claus and Boutilier 1998), where a_i is the action of selecting peer i as collaboration partner, r is the reward and λ is the learning rate (we use $\lambda = 0.9$). It would be natural to use the evaluation of the collaboration artifact created with peer i as the reward. Instead, an agent uses its evaluations of i's solitary artifacts as an approximation, learning how much it likes its peers artifacts. In our experience this works well, because the agent gets information about the peers from all created solitary artifacts and not just from its collaborations (Linkola and Hantula 2018).

The new learning scheme used in this paper, *ga-Q* (goal-aware-Q), extends hedonic-Q with goals. The update rule for ga-Q is $Q(g, a_i) \leftarrow Q(g, a_i) + \lambda(r - Q(g, a_i))$, where g is an aesthetic goal. For the reward ga-Q uses its own evaluation of its peers' artifacts just like hedonic-Q. Ga-Q learns how much it likes its peers' artifacts relative to g.

Ga-Q requires discrete goals, but the agent's aesthetic goal is a continuous value. We discretize the continuous value by dividing the range of possible goal values to B equal sized bins, resulting in B goals for ga-Q. So if the aesthetic goal is bounded in $[v_{min}, v_{max})$, this interval is divided into B, $(v_{max} - v_{min})/B$ sized subintervals, which represent the goals for ga-Q. From now on we refer to the discretized goals with g_b. We use $B = 20$.

One of the greatest benefits of ga-Q is, that the learning agent can update all possible goals simultaneously based on a single artifact. For each goal g_b in the ga-Q model, the artifact is evaluated using the middle point of the bin as the aesthetic goal. Then $Q(g_b, a_i)$ is updated using this evaluation as the reward. This way the agent already knows how to act, when it changes its aesthetic goal to a new one, even if it has never had that goal before.

An agent uses the peer model learned using hedonic-Q by sorting its peers into a preference order using their corresponding $Q(a_i)$ values. The ordering is done similarly with ga-Q, by first mapping the agent's current aesthetic goal to g_b and then using the $Q(g_b, a_i)$ values.

Adaptation during collaboration During collaboration, the agent that got to choose its partner in the collaboration pair uses the Q-values to choose a new aesthetic goal in the following way. If the selected partner is peer i, the temporary goal is the middle point of the bin that corresponds to the goal $\max_{g_b} Q(g_b, a_i)$. This means choosing the goal that maximizes the agent's appreciation of its collaboration partner's artifacts, maximizing the chance that the collaborating agents have some common ground, i.e. the agents appreciate similar artifacts. After the collaboration, the temporary goal is changed back to the goal the agent had before collaboration. The selected partner does not change its goal.

Strategic Movement

In this section, we describe how an agent strategically changes its aesthetic goal, i.e. moves. First, we describe how the movement is triggered, and then we describe how the new aesthetic goal is decided.

Choosing to move We model an agent's desire to move as a value c, accumulating whenever the agent observes an artifact. When c exceeds a fixed threshold c_t, an agent chooses a new aesthetic goal, i.e. moves. When an agent changes its goal, c is reset to 0.

An agent A accumulates c when observing artifact I as follows:

$$c = \begin{cases} c + \frac{1}{value(I)^n}, & \text{if } A \text{ is a (co-)creator of } I \\ c + \max\left\{0, 1 - \frac{|g-v|}{2s}\right\}, & \text{for other images,} \end{cases}$$

where n is the number of simulation steps since the agent's last goal change, g is the agents current aesthetic goal, v is the aesthetic measure value of the received artifact and s is the bin size used for ga-Q.

The formulation for (co-)created artifacts accumulates c exponentially faster the longer the agent fails to create value with its current goal. On the other hand, the accumulation is less pronounced when the agent has just moved, giving the agent time to adjust itself to its new aesthetic goal. The accumulation of c for other artifacts is larger when its peers are creating artifacts close to the agent's aesthetic goal, making the agent move in shorter intervals if its close to its peers.

The threshold c_t is designed so that an agent accumulates enough curiosity to trigger movement with every 10th time step under two assumptions. First, an agent is able to completely satisfy its own aesthetic goals (produces solitary and collaborated artifacts with value 1.0). Second, each observed peer artifact's objective aesthetic value is drawn from uniform distribution within the aesthetic bounds.

Moving Once the movement is triggered, an agent decides its new aesthetic goal based on its memory, the Q-values and potentially the agent's current aesthetic goal.

We describe two different ways an agent can strategically change its aesthetic goal: static and dynamic, both utilizing ga-Q. Static movement is as likely to move to any place within its aesthetic bounds. Dynamic movement prefers aesthetic goals closer to its current goal, decreasing the desire to move linearly with distance to a new aesthetic goal.

The new aesthetic goal is chosen as follows:

1. Agent calculates how many artifacts in its memory fall into each ga-Q goal bin, and filters out the four (20%) most crowded bins.
2. Agent filters out any remaining bins which are perceived to contain a peer, i.e. have an agent which has maximum Q-value in that bin, and estimates the collaboration potential of each remaining bin as the sum of the four highest Q-values in it.
3. If the movement type is dynamic, the agent scales each remaining bin's value according to its closeness to its current aesthetic goal.
4. Agent selects the bin with the highest value and chooses a new aesthetic target using a uniform distribution defined by the bin's borders.

The decision process is designed to satisfy an agent's novelty-seeking and collaboration goals. By filtering out the areas where most artifacts have been observed and the areas which are expected to contain peers, an agent tries to find a novel place within the society. The collaboration goal is satisfied by choosing a place which has a good collaboration potential.

Experiment Setup

With our experiments we aim to investigate intentional collaboration partner selection in a dynamic agent society, using goal-awareness to benefit the partner selection and the collaboration process, and finally guiding the aesthetic goal change with curiosity and goal-awareness. Our main research questions are:

1. How does the new learning scheme ga-Q perform? Especially compared to hedonic-Q.
2. How does adapting to the collaboration partner with a temporary aesthetic goal affect the results?
3. Does using curiosity and goal-awareness in changing one's aesthetic goal benefit collaboration?
4. What kind of emergent behavior arises on the society's level, when curiosity and goal-awareness are used in changing one's aesthetic goal?

In our experiments, we have 16 agents and 2 aesthetic measures. Half of the agents have entropy (ENT) as their aesthetic measure and the other half have fractal dimension (FRD). Each agent is initialized with a goal for the aesthetic measure. The aesthetic goal g for an agent A is initialized uniformly from the following bounds: $g \in [0.5, 4.5)$ if A uses entropy and $g \in [0.5, 1.8)$ if A uses fractal dimension.

The nucleus of our experiment setup is a simulation run consisting of 200 iterative time steps ($S = (s_1, s_2, \ldots, s_{200})$). The agents start each simulation with empty memories, creating the first images using only their aesthetics and evolutionary engine. The simulation is run 30 times for each learning scheme and movement configuration present in our experiments, resulting in a total of 180 runs. For the results we report the experiment setup run averages.

At odd time steps each agent creates a solitary artifact. At even time steps the agents select their collaboration partners and create artifacts in pairs. All of the created artifacts are sent to all agents for evaluation. An agent memorizes all of the artifacts it has created. An artifact created by another agent is memorized, if it exceeds the agent's thresholds for novelty (0.4) and value (0.5). If an agent changes its aesthetic goal, it does so at the start of an odd time step, before it starts creating a solitary artifact.

Table 1: Collaboration success ratios and various value measures for learning schemes. The statistics are averages of 30 simulation runs for each experiment configuration (column), displayed with 99% confidence interval.

	Collaborator selection and adaptation				Strategic movement	
Measurement	Hedonic-Q	Ga-Q_{fxd}	Ga-Q_{ada}	Random	Ga-Q_{st}	Ga-Q_{dyn}
Collab. success (CS) %	83.4 ± 0.8	83.5 ± 0.8	93.4 ± 0.5	67.4 ± 1.3	95.4 ± 0.6	94.7 ± 0.6
Value, own solitary	$.962 \pm .002$	$.962 \pm .002$	$.962 \pm .002$	$.960 \pm .001$	$.981 \pm .001$	$.969 \pm .005$
Value, collab. selector	$.934 \pm .002$	$.936 \pm .002$	$.939 \pm .001$*	$.892 \pm .003$	$.968 \pm .001$*	$.942 \pm .004$*
Value, collab. partner	$.941 \pm .002$	$.941 \pm .002$	$.946 \pm .002$	$.920 \pm .002$	$.974 \pm .001$	$.958 \pm .007$
Novelty, own solitary	$.536 \pm .004$	$.531 \pm .007$	$.532 \pm .004$	$.535 \pm .006$	$.595 \pm .006$	$.539 \pm .009$
Novelty, collab. selector	$.535 \pm .004$	$.528 \pm .007$	$.534 \pm .005$	$.523 \pm .006$	$.594 \pm .006$	$.541 \pm .009$
Novelty, collab. partner	$.534 \pm .004$	$.527 \pm .007$	$.533 \pm .005$	$.521 \pm .006$	$.594 \pm .006$	$.540 \pm .009$
–Collaboration between different aesthetics (selector $\rightarrow$ selected), collaboration attempt counts in brackets						
ENT $\rightarrow$ FRD CS%	83.5 ± 1.6 (4252)	83.9 ± 1.4 (4243)	86.8 ± 1.5 (4116)	70.6 ± 2.0 (6471)	91.8 ± 1.3 (4253)	89.0 ± 1.6 (3135)
FRD $\rightarrow$ ENT CS%	80.7 ± 1.6 (2680)	79.0 ± 2.2 (2703)	84.1 ± 2.5 (2632)	71.5 ± 2.8 (6381)	86.0 ± 2.3 (2523)	86.0 ± 3.0 (2599)
Total attempts	6932	6946	6748	12852	6776	5734

* *To make the values comparative between the experiments, the value is computed using the agent's real aesthetic goal, not the adapted goal for the duration of the collaboration.*

At the start of the collaboration time steps, the agents are arranged into a random order. Then one by one the agents use their preference list (defined by the learning scheme) to select a partner. The partner is the first agent in their preference list, which does not yet have a collaboration partner on this time step.

Creating a solitary and collaboration artifact takes roughly the same amount of resources. In our experiments, 10 evolutionary iterations are used for both solitary and collaboration artifacts. In the collaboration process, the agents do 5 iterations each.

Next, we describe our two different experiment setups.

Collaborator selection and adaptation With this setup we aim to investigate using goal-awareness for selecting collaboration partners in a dynamic situation and for adapting to the collaboration partner. These experiments also serve as a baseline for the strategic movement setup.

We run the setup for hedonic-Q and ga-Q (with and without adaptation), and compare the results to a baseline where the collaboration partners are selected randomly. We refer to these runs as hedonic-Q, ga-Q_{fxd} (without adaptation), ga-Q_{ada} (with adaptation) and random.

In this setup all learning schemes change their aesthetic goal randomly. The agents have 0.2 probability to change their goal in the beginning of each solitary time step, making them change their goal on average on every 10th time step. The new goal is drawn from uniform distribution in the bounds of the agent's aesthetic measurement (see above).

Strategic movement With this experiment we aim to see how using curiosity and goal-awareness in changing one's aesthetic goal affects the collaboration results and does strategic movement give rise to emergent phenomena on the macro-level.

We experiment with the two different strategic movements using ga-Q: static (ga-Q_{st}) and dynamic (ga-Q_{dyn}).

Results

We now proceed to present the results from our experiments.

Collaborator selection and adaptation We see from Table 1 that all peer modeling schemes are able to produce more collaboration artifacts (higher CS%) with higher value than random collaboration. Especially ga-Q $_{ada}$ is able to collaborate successfully, which means that adaptation to the partner's aesthetic goal is beneficial for our collaboration process. Surprisingly, there isn't much difference in CS% between hedonic-Q and ga-Q_{fxd}, although hedonic-Q has to relearn all Q-values every time the agent changes its aesthetic goal. Ga-Q_{fxd} should cope with the goal changes better, because it maintains Q-values for all goals simultaneously. We return to this in discussion.

Collaboration value Interestingly, adapting to the collaboration partner does not decrease the adapting agent's value for collaboration artifacts, as seen in row 3 of Table 1, comparing ga-Q_{fxd} to ga-Q_{ada} (the value is calculated using the agent's real aesthetic goal, instead of the temporary goal). This is probably caused by the selfish selection of partner by the selector agent, which is the one adapting, choosing peers who are close to its goal.

Collaboration between aesthetics From the last three rows of Table 1, it can be seen that hedonic-Q, ga-Q_{fxd} and ga-Q_{ada} all have significantly less collaboration between the aesthetics compared to random. Overall the entropy agents are able to select partners leading to more successful collaborations from the fractal dimension agents than vice versa. On rows 8 and 9 ga-Q_{ada} has statistically significantly higher CS% than ga-Q_{fxd} (Welch's t-test, p-values 4.5e-05 and 1.3e-04 respectively). This shows that adapting to the collaboration partner is beneficial for collaboration between the aesthetics.

Strategic movement In Table 1 the two rightmost columns show a general improvement in collaboration suc-

Table 2: Average moving distance, clustering and rate.

Measurement	Aesthetic	Random	Ga-Q_{st}	Ga-Q_{dyn}
Average area covered in 10 steps (normalized)	ENT	0.424	0.232	0.059
	FRD	0.416	0.198	0.048
Average number of agents in the same bin	ENT	1.217	1.925	1.104
	FRD	1.201	4.333	1.292
Average number of aesthetic goal changes	ENT	20.279	21.971	15.025
	FRD	19.533	27.313	17.217

cess and value for the strategic movement, compared to random movement.

Movement From Table 2 we observe that ga-Q_{st} agents change their aesthetic goal more and operate in a much larger aesthetic range within 10 steps than ga-Q_{dyn} agents. Still ga-Q_{st} has less overlap in the bins than ga-Q_{dyn}, indicating a less spread out society. This combined with the high values for ga-Q_{st} in Table 1, it seems that the ga-Q_{st} society is very opportunistic, always jumping to the most promising place together. Ga-Q_{dyn} is more conservative and spread out in its movement. We reflect on this more in discussion.

These differences between ga-Q_{st} and ga-Q_{dyn} can be seen in Figure 2, too. In the ga-Q_{st} runs the whole society moves tightly together, even when the target is oscillating intensely, as happens with entropy. With fractal dimension the society tends to stay in the high end of the aesthetic bounds. In the ga-Q_{dyn} runs the society also moves together, but in a more spread out manner. The collective targets of the society do not oscillate, but rather move steadily.

Novelty As seen in Table 1 rows 5-7, novelty is quite similar between all the schemes, except for Ga-Q_{st}. Ga-Q_{st} finds more novelty than the others due to its curiosity and ability to move in the whole aesthetic range. The high novelty is probably also partially caused by the FRD agents favoring complex artifacts, which tend to be more novel. However, we observed in our experiments, that entropy agents also produced notably more novelty with Ga-Q_{st} than the other schemes. Ga-Q_{dyn} is also guided by curiosity, but it mostly operates in a small range around its current target, making finding novelty more difficult.

Discussion and Conclusions

We have presented a new goal-aware peer model, ga-Q. The peer model enables an agent to envision alternative aesthetic goals, allowing the agent to temporarily adapt to its collaboration partner, and position its own aesthetic goals in relation to its peers' aesthetic goals. Our experiments indicate that the goal-aware selection of temporary goal for the collaboration is beneficial to our collaboration process and that the curious and goal-aware movement is beneficial for both collaboration and solitary artifact creation. Goal-awareness can also facilitate collaboration between the aesthetics.

Ga-Q Overall, ga-Q shows potential as a straightforward way to provide agents with a goal-aware peer modeling technique. It is easily generalizable to societies where new peers are introduced and old ones may leave. When a new peer

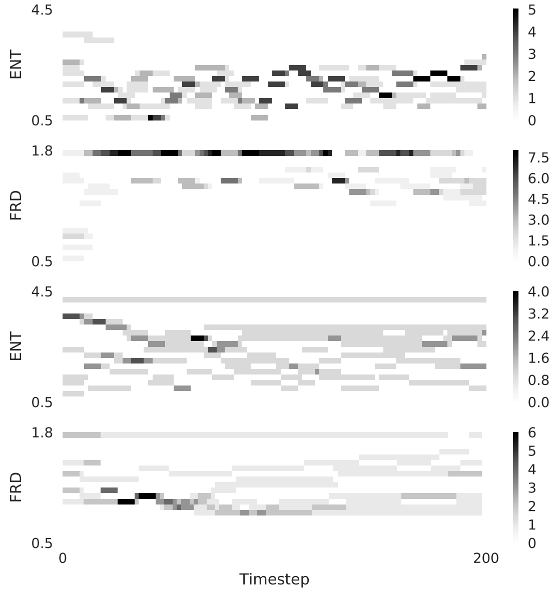


Figure 2: Heat maps of the whole society's typical aesthetic goal movement during a single run for static (upper two) and dynamic (lower two) strategic movement.

enters, a new Q-value $Q(g, a_j)$ can be created for each goal with a default value. When a peer leaves, all Q-values related to it can simply be dropped. Similarly, new goals can be created and old goals can be dropped. However, the number of goals in ga-Q grows exponentially with the number of an agent's aesthetic goals, making it impractical in situations where the number of aesthetic goals an agent has is high.

The way in which we use the ga-Q's Q-values to calculate the collaboration potential for a goal is quite unconventional. In our case, using the sum of the top Q-values makes sense, because an agent might not be able to select its favorite peer in the partner selection process, as that peer might already be in a collaboration pair. Therefore the agents should aim to select goals for which good collaboration partners exist, even if they don't get to select the best one.

Adapting to collaboration partner In the results, we observed that ga-Q without adapting to the collaboration partner is close to hedonic-Q. The reason is that the change of one's aesthetic goal happens before an agent receives the solitary artifacts from its peers. Q-learning's learning speed is fast enough to adapt using one step's worth of information. If the agent wouldn't get new information between its own aesthetic goal change and partner selection, or if the learning rate was lower, hedonic-Q wouldn't be able to make informed choices, while ga-Q should be relatively unaffected.

Strategic movement Our results for strategic movement are dividing. Even though ga-Q_{st} has the highest collaboration success, value and novelty, it might not be the most desirable way of implementing a society. The rapid nature of static movement's collective aesthetic goal changes (see Figure 2) renders the whole society unstable.

Ga-Q_{st} is also heavily affected by the asymmetric nature of the two aesthetic measures. The ENT agents produce nearly only artifacts which the FRD agents observe to belong to a couple of bins near the higher end of their aesthetic bounds. This causes the FRD agents to swarm around these bins, unable to move away from them.

Further, the ability to change one's aesthetic goal arbitrarily far might not be preferable, e.g. agents drastically changing their aesthetic goal might not be able to make full use of their accumulated expertise. For a more spread out and conservative search of the domain, ga-Q_{dyn} seems preferable. However, the two different strategic movements can be seen as different points on the same scale: how much the agent prefers new aesthetic goals close to its current aesthetic goal.

Lastly, memorylessness of our strategic movement implementation makes it undesirable for long processes. The agent does not accumulate information of the aesthetic goals it has previously possessed, and thus the swarming behavior of the static FRD agents may emerge. For more sustained processes, time-awareness has to accompany strategic movement in order for the agent to understand the history of its own aesthetic goals and utilize that knowledge in its decision making.

To conclude, we believe that for true social intent, the agents need to model their peers and their interaction. By experiments we hope to have gathered some insight towards such intent. In the future, we aim to study more closely how time-awareness can be used in strategic movement in conjunction with goal-awareness and how societies with diverse strategic movement behaviors evolve over time.

Acknowledgments. This work has been supported by the Academy of Finland under grant 313973 (CACS).

References

Boden, M. 1992. *The Creative Mind*. London: Abacus.

Castelfranchi, C. 1998. Modelling social action for AI agents. *Artificial Intelligence* 103(1):157–182.

Claus, C., and Boutilier, C. 1998. The dynamics of reinforcement learning in cooperative multiagent systems. In *Proceedings of the Fifteenth National/Tenth Conference on Artificial Intelligence/Innovative Applications of Artificial Intelligence*, 746–752.

Csikszentmihalyi, M. 1988. Society, culture, and person: A systems view of creativity. In Sternberg, R. J., ed., *The Nature of Creativity: Contemporary Psychological Perspectives*. Cambridge University Press. 325–339.

den Heijer, E., and Eiben, A. 2014. Investigating aesthetic measures for unsupervised evolutionary art. *Swarm and Evolutionary Computation* 16:52–68.

Eigenfeldt, A.; Bown, O.; Brown, A. R.; and Gifford, T. 2017. Distributed musical decision-making in an ensemble of musebots: Dramatic changes and endings. In *Proceedings of the Eight International Conference on Computational Creativity*, 88–95.

Gabora, L., and Tseng, S. 2014. The social impact of self-regulated creativity on the evolution of simple versus complex creative ideas. In *Proceedings of the Fifth International Conference on Computational Creativity*, 8–15. Ljubljana, Slovenia: Josef Stefan Institute, Ljubljana, Slovenia.

Jennings, K. E. 2010. Developing creativity: Artificial barriers in artificial intelligence. *Minds and Machines* 20(4):489–501.

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.

Lewis, P. R.; Chandra, A.; Faniyi, F.; Glette, K.; Chen, T.; Bahsoon, R.; Torresen, J.; and Yao, X. 2015. Architectural aspects of self-aware and self-expressive computing systems: From psychology to engineering. *Computer* 48(8):62–70.

Linkola, S., and Hantula, O. 2018. On collaborator selection in creative agent societies: An evolutionary art case study. In *Proceedings for the Seventh International Conference on Computational Intelligence in Music, Sound, Art and Design*, 206–222.

Linkola, S.; Kantosalo, A.; Männistö, T.; and Toivonen, H. 2017. Aspects of self-awareness: An anatomy of metacreative systems. In *Proceedings of the Eight International Conference on Computational Creativity*, 189–196.

Paulus, P., and Nijstad, B. 2003. *Group Creativity: Innovation through Collaboration*. Oxford University Press.

Pérez y Pérez, R.; Negrete, S.; Nalosa, E. P.; Ávila, R.; Castellanos, V.; and Lemaitre, C. 2010. Mexica-impro: A computational model for narrative improvisation. In *Proceedings of the International Conference on Computational Creativity*, 90–99.

Romero, J., and Machado, P., eds. 2007. *The Art of Artificial Evolution: A Handbook on Evolutionary Art and Music*. Natural Computing Series. Springer Berlin Heidelberg.

Salehie, M., and Tahvildari, L. 2009. Self-adaptive software: Landscape and research challenges. *ACM Transactions on Autonomous and Adaptive Systems* 4(2):14:1–14:42.

Saunders, R., and Bown, O. 2015. Computational social creativity. *Artificial Life* 21(3):366–378.

Saunders, R., and Gero, J. S. 2001. Artificial creativity: A synthetic approach to the study of creative behaviour. In *Computational and Cognitive Models of Creative Design V*, 113–139. Sydney, Australia: University of Sydney.

Sims, K. 1991. Artificial evolution for computer graphics. In *Proceedings of the 18th Annual Conference on Computer Graphics and Interactive Techniques*, 319–328.

Sosa, R., and Gero, J. S. 2005. Social models of creativity. In *Proceedings of the International Conference of Computational and Cognitive Models of Creative Design VI*, 19–44. Heron Island, Australia: Key Centre of Design Computing and Cognition, University of Sydney, Australia.

Uzzi, B., and Spiro, J. 2005. Collaboration and creativity: The small world problem. *American Journal of Sociology* 111(2):447–504.

Watkins, C. J., and Dayan, P. 1992. Q-learning. *Machine learning* 8(3-4):279–292.

eRebuild Level Creator
TECHNOLOGY AND ARTS SHOW AND TELL

Dan Smith
Computational Science Department
Florida State University
Tallahassee, FL 32306 USA
das10c@my.fsu.edu

Abstract

eRebuild is a learning game initially created as a learning tool that integrates data-based assessment. As the game develops into a robust learning platform, more content becomes necessary. To keep the content focused on learning, we enlist math teachers as level designers. By creating a mixed-initiative co-creative environment, even the complete novice level designer can create a targeted learning experience.

Introduction

It is frequent repetition that produces a natural tendency. (Aristotle)

Nowhere is this more apparent than the field of mathematics. This repetition can be seen easily in how we teach math to children. Worksheets, flash cards, multiplication table drills, and even most math-based video games embody the *practice makes perfect* mantra.

During the development of eRebuild, a math-learning based game, repetition had to be temporarily placed on the back burner. While developing tasks, the game focused on assessment, which necessitated breadth not depth. In addition, the game was intended to be different from early learning games. We are not creating digital flashcards. Instead, game design and math content should support each other, not limit one another.

As eRebuild has grown into a learning platform in addition to an assessment tool, the demand for more levels has become apparent. The time required to create these levels manually is equally apparent. Procedural tools can generate generic levels that mimic those already seen, but if these systems produce a level of low quality, education suffers and the experience can feel much like the repetitive drill eRebuild was designed to avoid.

Each level should be novel in its presentation, and each task must have value as a math-learning tool. As such, generated levels must offer both of these properties. eRebuild's initial procedural generation systems lack assessment for both of these parameters (novelty, value).

As such a Mixed-Initiative co-creative environment has been created. Domain experts ensure that each level highlights its focal task and adjust the play area to ensure a balanced difficulty with minimal distractors.

eRebuild

Created by an interdisciplinary team of mathematicians, assessment experts, math educators, and architects, eRebuild is a math-learning game for students in grades six through eight. The game began development as a learning tool that integrates the data-based assessment, scoring student gameplay data as proficiency in a number of the middle school common core state standards. Players navigates through a 3D world recovering from a recent disaster rebuilding homes and schools using their unique abilities. They also collect and trade construction materials, and allocate spaces to displaced people. Each level was designed to elicit evidence of learner ability in the focal competency.

As one transitions to a more robust learning tool, additional content becomes necessary to enable task repetition. This content is displayed through game levels. To mitigate the time cost of creating these new levels, eRebuild includes a co-creative level editor aimed at teachers as opposed to game designers.

When designing a level, the creator chooses one or more target competencies from a list of supported math standards, based on which the level editor recommends a number of task types. The level is then procedurally generated based on the tasks selected. If the level generated is unsatisfactory, the generation process can be repeated . Once the level is complete, the teacher can further adjust parameters such as number, level locations, and type of subgoals regenerating the level as necessary. Finally, narrative elements can be add ed to selected items within the scene.

Conclusion

Unfortunately, the time required to create quality games with a variety of content remains large. Procedural content generation is one way to resolve this issue. Many of these methods leave something to be desired in the way of creativity.

By combining the procedural system with a level editor, the sum of the two becomes greater than the individual parts. A co-creative environment allows for a quicker development time than using the editor alone and gives the procedural system, a much needed human touch and some fine tuning.

Acknowledgement

This material is based upon work supported by the National Science Foundation under Grant No.1720533.

How Shell and Horn make a Unicorn: Experimenting with Visual Blending in Emoji

João M. Cunha, Pedro Martins, Penousal Machado
CISUC, Department of Informatics Engineering
University of Coimbra
{jmacunha,pjmm,machado}@dei.uc.pt

Abstract

Emoji are becoming increasingly popular, both among users and brands. Their impact is such that some authors even mention a possible language shift towards visuality. We present a Visual Blending-based system for emoji generation, which is capable of representing concepts introduced by the user. Our approach combines data from ConceptNet, EmojiNet and Twitter's Twemoji datasets to explore Visual Blending in emoji generation. In order to assess the quality of the system, a user study was conducted. The experimental results show that the system is able to produce new emoji that represent the concepts introduced. According to the participants, the blends are not only visually appealing but also unexpected.

Introduction

The word *emoji* has a Japanese origin, in which the *e* means "picture", *mo* means "writing" and *ji* means "character"[1] – leading to the often attributed meaning "picture-word". Emoji seems to have become an important part of our way of writing. Their increasing usage is well documented by the importance given to them by language related resources – Oxford Dictionaries named the emoji "Face With Tears of Joy" the Word of The Year of 2015[2] – and by statistical data – Facebook reported in 2017 that 60 million emoji are used every day on Facebook and 5 billion on Messenger[3].

Some authors even discuss a shift towards a more visual language (Lebduska 2014; Danesi 2017). This shift would in fact bring us close to old ways of writing, such as hieroglyphs. Using images as complementary signs in written communication enriches it (Niediek 2016) by allowing the transmission of non-verbal cues (e.g. face expressions, tones and gestures) (Hu et al. 2017), which are lacking in written communication and Computer-Mediated Communication (CMC). This integration in written language is easy to observe when we consider the increasing number of emoji-related tools and features. Some examples are Search-by-emoji supported by Bing[4] and Google[5], and the Emoji Replacement and Prediction features available in iOS 10[6]. We believe that other possible applications exist, specially in the domain of image generation (see some examples in Fig. 1).

Figure 1: Examples of visual blends. From left to right and top to bottom: *man apple*, *apple man*, *world peace*, *hot dog*, *rain cat*, *woman wonder*, *wonder woman*, *man bat*, *dinosaur park*, and *true blood*

Before emoji, sequences of ASCII characters were often used to express emotions CMC – emoticons (see Fig. 2). Despite the high adoption of emoji, some emoticons still continue to be used as an alternative due to their potential for customisation (Guibon, Ochs, and Bellot 2016). Whereas emoticons are composed of individual and replaceable parts, emoji are inserted as a whole in the text (Dürscheid and Siever 2017). In 2015, "skin tone" modifiers were added to Unicode core specifications and in 2016 the Unicode Consortium decided to implement the ZWJ (Zero-Width-Joiner) mechanism – an invisible character to denote the combination between two characters (Abbing, Pierrot, and Snelting 2017). This meant that new emoji could be created through the combination of existing ones, without the need to go through the Unicode Consortium.

Having the modifiers, the ZWJ mechanism and emoticons' combinational character as inspiration, it is our belief that Visual Blending can be explored to further extend emoji system. Visual Blending, which draws inspiration from Conceptual Blending (CB) theory (Fauconnier and Turner

[1] unicode.org/reports/tr51/proposed.html, retr. 2018
[2] en.oxforddictionaries.com/word-of-the-year/word-of-the-year-2015,retr. 2018.
[3] blog.emojipedia.org/5-billion-emojis-sent-daily-on-messenger/, retr. 2018.
[4] blogs.bing.com/search/2014/10/27/do-you-speak-emoji-bing-does, retr. 2018.
[5] forbes.com/sites/jaysondemers/2017/06/01/could-emoji-searches-and-emoji-seo-become-a-trend/, retr. 2018.
[6] macrumors.com/how-to/ios-10-messages-emoji/, retr. 2018.

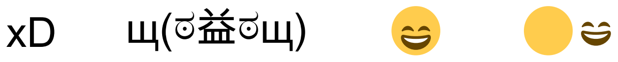


Figure 2: “Laughing” western emoticon, “Why” eastern emoticon, “Grinning Face With Smiling Eyes” emoji, and deconstruction of an emoji

2002), is a Computational Creativity (CC) technique which consists in merging two or more visual representations (e.g. images) to produce creative visual artifacts.

We propose a system based on Visual Blending and Semantic Network exploration to generate visual representations for introduced concepts (see Fig. 1). The blending process combines existing emoji to create novel ones. The results obtained vary in terms of conceptual complexity, going from literal to metaphoric. We believe that our approach has potential to be explored as an ideation-aiding tool to be used in brainstorming activities, presenting the user with representations for introduced concepts. With this goal in mind, in this paper we value creative and unexpected results and give less importance to literal and unambiguous ones (normally valued in emoji). We present the results of a user study focused on two-word concepts, which analyses the system output in terms of representation quality and surprise degree.

Related Work

Our work addresses two different topics: *Emoji* and *Visual Blending*. As such, we will firstly describe the state of the art for the two topics and then present projects or products which are related to Variation and Customisation of emoji.

Research on Emoji

Previous research on emoji can be mostly divided into the following categories: Meaning, Sentiment, Interpretation, Role in communication, and Similarity between emoji.

Studies on emoji meaning often use word embedding techniques and different data sources (Dimson 2015; Barbieri, Ronzano, and Saggion 2016; Eisner et al. 2016).

In terms of research on emoji sentiment, Novak et al. (2015) provided the first emoji sentiment lexicon, and Hu et al. (2017) compared the sentiments of emoji to the overall sentiment of the message where they occur.

Miller et al. (2016) studied how the users' interpretation of meaning and sentiment of emoji change within and across-platforms, and Rodrigues et al. (2018) addressed how it may differ from intended meanings of developers and researchers.

Some authors address the role of emoji in written communication: Donato and Paggio (2017) studied emoji redundancy and part-of-speech category; Dürscheid and Siever (2017) discussed the function of emoji (complement vs replace); Gustafsson (2017) presented evidence that using emoji to replace words increases reading time; and Wicke (2017) investigated whether emoji could be seen as semantic primes.

Ai (2017) semantically measured emoji similarity. Other authors identified clusters of similar emoji based on emoji vector embeddings (Eisner et al. 2016; Barbieri, Ronzano, and Saggion 2016). Pohl et al. (2017) used a relatedness-hierarchy to organise emoji. Wijeratne et al. (2017b) created a dataset which contains human-annotated semantic similarity scores assigned to emoji pairs.

On emoji generation, few research work has been conducted and it will be addressed in a later section.

Visual Blending

Visual Blending consists in merging two or more visual representations (e.g. images) to produce new ones. In the context of CC, it is often used together with CB methods to produce representations for a blended mental space. In such cases, it is called Visual Conceptual Blending.

One of the earliest attempts to computationally produce visual blends is, to the best of our knowledge, The Boat-House Visual Blending Experience (Pereira and Cardoso 2002). The work resulted from experiments in interpretation and visualisation of conceptual blends produced for the input spaces *house* and *boat* (Goguen 1999) by an initial version of Divago – one of the first artificial creative systems based on CB theory (Pereira 2007). The visual representations were drawn using a Logo-like programming language.

Ribeiro et al. (2003) used a 3D interpreter to visualise blends of novel creatures produced by Divago from a set of existing ones. The concept maps provided by Divago were converted by the interpreter into Wavefront OBJ files, which could then be rendered.

Steinbrück (2013) presented a framework aimed at exploring the application of CB to the visual domain. It combines image processing techniques with semantic knowledge gathering to produce images in which elements are replaced with similar-shaped ones (e.g. round medical tablets are transformed into globes).

Confalonieri et al. (2015) proposed the use of argumentation to evaluate and iteratively refine the quality of blended computer icons. The authors introduced a semiotic system, which was based on the idea that signs can be combined to convey multiple intended meanings. Despite this, no evidence of a possible implementation was provided.

Xiao and Linkola (2015) presented Vismantic, a semi-automatic system which uses three binary image operations (juxtaposition, replacement and fusion) to produce visual compositions for specific meanings (e.g. *Electricity is green* is represented as the fusion between an image of an electric light bulb with an image of green leaves). The intervention of the user is necessary for both the selection of images and the application of the visual operations.

Correia et al. (2016) developed X-Faces as an approach to Data Augmentation for Face Detection purposes. The system autonomously generates new faces out of existing ones by recombining face parts (e.g. eyes, nose or mouth), using evolutionary algorithms and computer vision techniques.

Cunha et al. (2017) proposed a system for automatic generation of visual blends using a descriptive approach. It used structured representations along with sets of visual relations which describe how the parts – in which the visual representation can be decomposed – relate among each other.

The potential of deep neural networks in tasks related to visual blending has been pointed out by several authors (Berov and Kuhnberger 2016; McCaig, DiPaola, and Gabora 2016; Heath and Ventura 2016). One example is the work DeepStyle (Gatys, Ecker, and Bethge 2015), which explores style transfer in image rendering by recombining the content of an arbitrary image with a given rendering style (e.g. painting styles).

In terms of character blending, one example is the blend of Pokémon (both image and name)[7]. On the same subject, Liapis (2018) produces mappings between type and attributes (e.g. color, shape and in-game sprite), which allow the change of type of a Pokémon.

Current computational approaches to visual blending can be divided into two groups in terms of type of rendering used: the ones which attempt to blend pictures or photorealistic renderings; and the ones that focus on non-photorealistic representations, such as pictograms or icons.

On the other hand, a categorisation can also be done in terms of where the blending process occurs: some interpret or visualise previously produced conceptual blends – e.g. Pereira and Cardoso (2002); others use blending only at the visual level – e.g. Correia et al. (2016); and in others, which can be called hybrid, the blending process starts at the conceptual level and only ends at the visual level – e.g. Cunha et al. (2017).

Variation, Customisation and Generation

Despite the emoji lexicon being constantly increased, there are still a large number of concepts which have not yet found their way into emoji. This is especially evident for more abstract concepts which do not meet the criteria established in the Unicode Guidelines for new emoji. However, several attempts have still been made to complement the system, e.g. *sleep working* by Mentos[8] and *drop the mic* by Microsoft[9]. This shows that the visual representation of more abstract, ambiguous concepts is also valued by the general public.

There are also several examples of user customisation. Windows Live Messenger[10] allowed the user to create emoticons by uploading an image file and Slack[11] currently has the same feature. Some applications allow face-related customisation, e.g. Bitmoji[12], and Taigman, Polyak and Wolf (2016) transform photos of faces into cartoons.

All these examples, serve to show that there is great potential in emoji variation, customisation, and, above all, generation. Despite this, few research work has been conducted on the topic. One example which is related to variation is Barbieri et al. (2017), which investigated the properties of derivations of the kappa emote in Twitch. Specific research on emoji generation mostly uses Generative Adversarial Networks to replicate existing emoji, e.g. (Puyat 2017; Radpour and Bheda 2017). The work of Radpour and Bheda (2017) is particularly interesting, as it is closely related to the idea of our paper by presenting some results for emoji blends. The quality of the results is, however, significantly lower than the one of official emoji, due to visual noise.

The closest work to ours is Emojimoji[13], an emoji generator implemented as part of the Emblemmatic project which also uses Twemoji. It randomly merges emoji shapes and names. However, none of the aforementioned examples uses semantic knowledge in emoji generation, which is the focus of our work.

Figure 3: Visual blends for *rain man* using the same emoji. The first uses juxtaposition and the others use replacement.

The Approach

Current needs for more variation and customisation serve as support and inspiration to our main goal: the development of a system that visually represents concepts introduced by the user. This system can be used for several purposes, among which aiding in ideation processes or generating new emoji. Our approach combines data from ConceptNet (Speer and Havasi 2012), EmojiNet (Wijeratne et al. 2017a) and Twitter's Twemoji[14] dataset to explore Visual Blending of emoji.

Resources used

As already mentioned, several resources are put together when developing this system:

- Twitter's Twemoji: a fully scalable vector graphics dataset made available by Twitter. This dataset only consists of images without any semantic information besides the corresponding unicode in the name of each image file. The version used is Twemoji 2.3, which has 2661 emoji;
- EmojiNet: a machine readable sense inventory for emoji built through the aggregation of emoji explanations from multiple sources (Wijeratne et al. 2017a). It was used to provide semantic knowledge to the emoji of the Twemoji dataset despite only having data regarding 2389 emoji;
- ConceptNet: a semantic network originated from the project Open Mind Common Sense (Speer and Havasi 2012). It is used to get concepts related to the one introduced by the user.

The decision to use fully scalable vector graphics is aligned with some of our previous work (Cunha et al. 2017). This image format enables scaling without reducing quality and uses a layered structure – each part of an emoji (e.g. a mouth) is in a separate layer (see Fig. 2). This structure allows an easier blending process and contributes to the overall sense of cohesion among the parts.

[7]pokemon.alexonsager.net, retr. 2018

[8]ementicons.mentos.com/en_GB, retr. 2018

[9]http://huffingtonpost.com/visualnewscom/neil-degrasse-tyson-and-4_b_5615887.html, retr. 2018

[10]news.microsoft.com/2003/06/18/msn-messenger-6-allows-im-lovers-to-express-themselves-with-style/,retr. 2018

[11]get.slack.help/hc/en-us/articles/206870177-Create-custom-emoji, retr. 2018

[12]bitmoji.com, retr. 2018

[13]emblemmatic.org/emojimoji, retr. 2018

[14]github.com/twitter/twemoji, retr. 2018

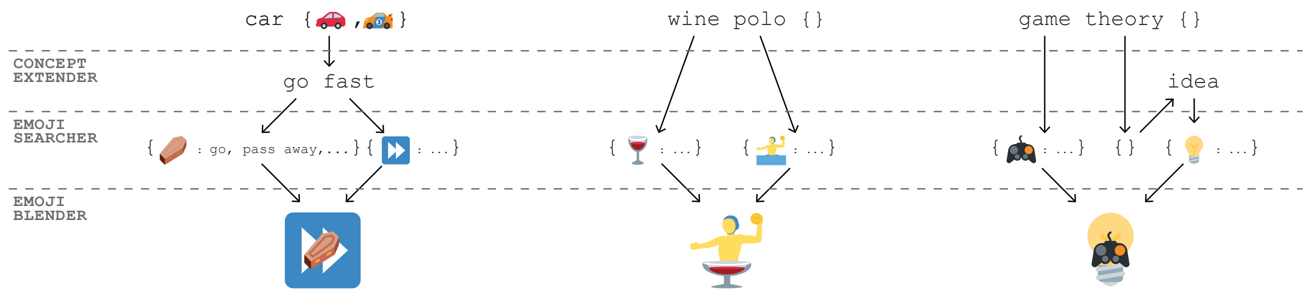


Figure 4: Generation of visual representations (T2) for three concepts: *car*, *wine polo* and *game theory*

In terms of the semantic knowledge, we initially used the emoji name, emoji definition, emoji keywords and sense definitions – all provided by EmojiNet. However, we concluded that using sense descriptions often leads to unrelated or too specific emoji, which are not useful for the system. For this reason, we decided to use the sense lemmas (word(s) that identify the sense) instead of their descriptions. Unfortunately, the EmojiNet dataset only includes the sense id and its descriptions. In order to solve this problem, the lemmas for each sense id were gathered from BabelNet (Navigli and Ponzetto 2012), which was the original source of the EmojiNet sense data (Wijeratne et al. 2017a).

General Architecture

The system searches existing emoji semantically related to the introduced concept and complements this search with a visual blending process which generates new emoji. In a ideation process, the blending process is useful when there is no existing emoji that matches the concept but also to suggest possible alternatives.

The system consists of two main tasks – retrieval of existing emoji that match the introduced concept (T1) and generation of new ones through visual blending (T2) – which are conducted using three components:

1. Concept Extender (CE): searches ConceptNet for related concepts to the one introduced;
2. Emoji Searcher (ES): searches emoji based on words given, using semantic data provided by EmojiNet;
3. Emoji Blender (EB): receives two emoji as input and returns a list of possible blends.

The system output is a set of visual representations for the introduced concept, composed of existing emoji and generated blends. The system produces a variable number of visual blends, depending on the data found (e.g. Fig. 3).

How it works

The current version works with concepts composed of a maximum of two words. The system starts by analysing the text given by the user. In this first stage, three things can happen: (i) the user introduces a single word (e.g. *car*), (ii) two words (e.g. *wine polo* or *game theory*) or (iii) more. In the last case, the system removes stop-words (e.g. "a", "because", "before", "being", etc.) and considers the result as input text – if after these removal, the word count is still higher than two, the system ignores it and ends the process without any result.

Retrieval of Existing Emoji (T1) In order to conduct T1, the system mainly makes use of the *Emoji Searcher (ES)* component, which uses EmojiNet dataset to find emoji based on the word(s) given by the user (e.g. in Fig. 4 the *coffin* emoji is retrieved for the word *go* due to its presence in the sense "go, pass away,..."). The word searching is conducted in different places: emoji name and definition, keywords associated with the emoji and senses related to it.

The matching score – i.e. how well an emoji matches the word(s) – is calculated based on the results of the semantic search and the unicode codepoint length ("U+1f474" is more specific than "U+1f474 U+1f3fb"). A value is assigned to each of the criteria:

Name (NV): number of (#) words that match the word(s) searched divided by the total # words in emoji name;

Definition (DV): # words that match the word(s) searched divided by the total # words in emoji definition;

Keywords (KV): (1-1/(# matching keywords))×0.5 + ((# matching keywords)/(total # keywords))×0.5;

Sense (SV): (1-1/(# matching senses))×0.5 + ((# matching senses)/(total # senses))×0.5;

Unicode Codepoint (UV): 1/Codepoint length.

In order to produce the final matching score, the individual values are used together. The criteria have different weights due to importance of each one (e.g. a word in the name is more important than in a sense). Moreover, name, keywords and description were initially gathered from the Unicode Consortium, whereas senses were based on user attribution and may be more ambiguous. The criteria are then weighted according to the following formula: *Emoji matching value* = KV×0.3+NV×0.3+SV×0.2+DV×0.15+UV×0.05

After the searching process is concluded, the system produces a list of emoji that are related to the word given by the user, sorted by *emoji matching value* (e.g. the red and orange cars for the concept *car* in Fig. 4).

Generation of visual representations (T2) In T2 the system behaves differently, depending on the number of introduced words. In the case of single-word concepts, the blending between emoji of the same word does not occur, e.g. two existing emoji for *car* (the red and orange in Fig. 4) are not blended together to represent the concept *car*. This would only happen if the concept introduced was "car car". Instead, the *Concept Extender* and the *Emoji Searcher* components are used to get the emoji to blend.

The Concept Extender (CE) component is used to query ConceptNet for a given word, obtaining related concepts, sorted according to ConceptNet weight system. In the case of single-word introduced concepts, we only consider two-word related concepts (e.g. *go fast* in Fig. 4) as initial experiments indicated that using emoji from two single-word related concepts would result in blends unrelated to the introduced concept. After obtaining the two-word related concepts, the *ES* component (already described for T1) searches for emoji for each word (e.g. in Fig. 4 the *coffin* emoji is obtained for *go*, and the *fast forward* for *fast*). These emoji are then used in the blending process.

On the other hand, when the user introduces a two-word concept, the system firstly searches for existing emoji for each word, using the *ES* component (already described). If emoji are found for both words (e.g. *wine glass* emoji for *wine* and *polo player* for *polo* in Fig. 4), a process of blend is conducted. If the system does not find existing emoji for both words, a search for related concepts is performed, using *CE* component (already described). An example is shown in Fig. 4, in which no emoji is found for *theory*. The system uses the *CE* component to obtain related concepts (e.g. *idea*). After getting the related concepts, the system uses *ES* to search for matching emoji (e.g. *light bulb*). If the search is successful, a blending process is conducted.

The *Emoji Blender (EB)* component is where the blending process occurs, which consists in merging two emoji. The base emoji are selected from the retrieved lists provided by *ES*. In terms of blending, we consider three different methods, even though only two of them are currently being used – these are similar to the ones used in Vismantic (Xiao and Linkola 2015), initially inspired by Phillips and McQuarrie (2004). The first method is *Juxtaposition*, in which the two emoji are put side by side or one over the other (e.g. the blends for *car* and *game theory* in Fig. 4). The second method is *Replacement*, in which part of emoji *A* is replaced by emoji *B* (e.g. in the blend for *wine polo* the water is replaced by wine, see Fig. 4). A blend is produced for each part of emoji *A*: emoji *B* replaces the part using its position (e.g. in Fig. 3 the *rain cloud* emoji replaces the "moustache", the "face shape", the "hair", and the "nose"). The third method is *Fusion*, in which the two emoji are merged together by exchange of individual parts (not used in this paper).

Results and Discussion

In this section we present and discuss the experimental results. We begin by describing an user study and its results. Then, a general analysis of the system and the generated blends is made. Afterwards, we compare the system with previous work, addressing its strengths and shortcomings. In this paper, our goal is to focus on the generation of new visual representations and, for this reason, few attention is given to the process of existing emoji retrieval. In addition, we decided to limit our discussion and evaluation to two-word concepts, following our line of research on visual conceptual blending (Cunha et al. 2017). We intend to address single-word concepts in the future.

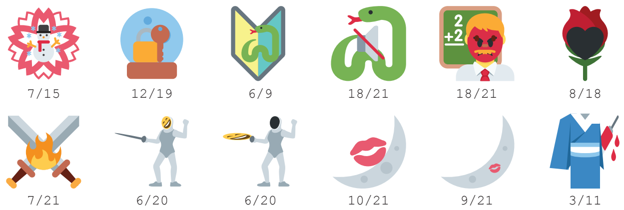


Figure 5: Blends selected as best representation for each concept (top 12). Below each blend is the number of participants who selected it and the total number of participants who selected a blend for that concept. The blends are ordered left-right, top-bottom, according to the order used in Table 1. Two blends are shown for *The Laughing Blade* and *The Sexy Moon*.

Evaluating results

In order to assess the quality of system in terms of blend production, a study with 22 participants was conducted. The main goal was to present the participants with blends and ask them to answer a series of questions related to blend quality.

Firstly, a list of ten concepts was produced. These were randomly generated on the website *Title Generator*[15]. The ten concepts are: *Frozen Flower*, *Secrets in the Future*, *Serpent of the Year*, *Silent Snake*, *Storm of the Teacher*, *The Darkest Rose*, *The Flame of the Swords*, *The Laughing Blade*, *The Sexy Moon*, and *The Sharp Silk*. The blends produced by the system for these concepts were shown to the participants. It is important to mention that the number of blends generated is variable and, consequently, the quantity of blends shown was not the same for every concept (e.g. *Silent Snake* has 7 blends and *Storm of the Teacher* has 47).

Each participant saw the blends of every concept but the order in which these were seen was not the same – this was done to minimise the biasing of the results. For each concept, the participants were asked to execute the following tasks: T1 – introduce the concept and generate the blends (presented all at once, side by side); T2 – answer if there is a blend that represents the concept (yes or no); T3 – evaluate quality of representation from 1 (very bad) to 5 (very good); T4 – identify degree of surprise from 1 (very low) to 5 (very high); T5 – select the best blend (only if a positive answer was given to T2). A section for the participants to write optional comments was also included. Asking the user to select the best blend and then make an evaluation of the system based on it may not be the proper way to conduct a user study. However, in the case of our system, it serves the purpose as the end goal is to use it in a process of ideation, in which having at least one good solution is enough.

The results obtained are shown in Tables 1 and 2. Overall, the system was able to generate blends that represented the

[15] ruggenberg.nl/titels.html, retr. 2018

Table 1: Number of answers to T2, T3 and T4

Concepts	**T2 (represented)** Yes	No	**T3 (quality)** <4	≥4	**T4 (surprise)** <4	≥4
Frozen Flower	13	9	6	7	7	6
Secrets in the Future	18	4	7	11	8	10
Serpent of the Year	5	17	1	4	1	4
Silent Snake	20	2	7	13	7	13
Storm of the Teacher	20	2	5	15	4	16
Darkest Rose	18	4	4	14	10	8
The Flame of the Swords	21	1	3	18	13	8
The Laughing Blade	16	6	12	4	7	9
The Sexy Moon	21	1	5	16	6	15
The Sharp Silk	5	17	4	1	2	3
Total	157	63	54	103	65	92
General Mode			4		4	
General Median			4		4	

Table 2: Mode and median for T3 and T4 (only includes participants who answered positively to T2)

Concepts	**T3 (quality)** Mode	Median	**T4 (surprise)** Mode	Median
Frozen Flower	4	4	3	3
Secrets in the Future	4	4	4	4
Serpent of the Year	4	4	4	4
Silent Snake	4	4	4	4
Storm of the Teacher	4	4	4	4
The Darkest Rose	4	4	3	3
The Flame of the Swords	4	4	3	3
The Laughing Blade	3	3	3	4
The Sexy Moon	4	4	4	4
The Sharp Silk	3	3	2 and 5	4

concepts – 71.36% (157 out 220) of the answers to T2 were positive (see Table 1) and the quality was above or equal to high (4) in 46.81% (103 out of 220) of the cases.

Moreover, the system is able to produce different blends which can be considered interesting for the same concept. For example, two blends are shown for *The Laughing Blade*, which were selected as the best by the same number of participants (Fig. 5). One reason for this may be the different interpretations for *The Laughing Blade*: a metaphor for the name of the swordsman; or the blade is literally laughing. Similarly, the best blend for *Storm of the Teacher* is metaphoric and for *The Flame of the Swords* is literal. The surprise results seem to reflect this difference: *The Flame of the Swords*, despite having good quality score, was not considered surprising by the majority of the participants, whereas *Storm of the Teacher* was considered both surprising and of good quality.

The worst results were the ones from *The Sharp Silk*, which was only considered concept-representative by 5 participants, from which only one assigned a quality score above or equal to high (4). Their opinion on the surprise criterion was also divided, resulting in two modes (2 and 5).

Most participants reported having difficulty in understanding some of the blends. Some did not recognise a shape

Figure 6: Generation issues. Three blends for *dog* using different related concepts (*drink water*, *guard house*, and *sense danger*, on the left), and blends for *cold* and *unicorn*

(e.g. red shape of *Frozen Flower*), others had different interpretations (a planet instead of a crystal ball for *Secrets in the Future*) and others did not understand the reason behind a blend (e.g. *Serpent of the year*) – see Fig. 5. These were the main reasons for answering negatively to T2, and possibly for the difference in the participants opinion.

General Analysis

Overall, we consider that the results obtained are visually and conceptually interesting (even though no conceptual blending is performed) and, in most cases, unexpected which is supported by the results obtained in the user study.

The system is able to generate variable results, both with the same emoji – e.g. *rain man* in Fig. 3 – and with different ones – e.g. *dog* in Fig. 6. The blending process, through the use of Juxtaposition and Replacement, produces blends that represent the concept behind them and vary in terms of degree of conceptual complexity – in Fig. 1 the blend for *hot dog* is harder to understand than the one for *man bat*. Moreover, the system is able to make less direct connections, e.g. *wine polo* has a literal representation whereas the one for *car* is metaphoric (Fig. 4).

There is no doubt that the performance of the system is dependent on the input emoji and the semantic knowledge associated with it. As such, it might generate interesting blends for some concepts and uninteresting for others. Moreover, in the current implementation, only the emoji with highest matching value is used – changing this would increase the number of resulting visual blends and possibly lead to the generation of better ones (the highest matching value does not necessarily result in the best blends).

The results depend on the word order. The blends generated differ depending on the order of the words introduced. Examples of this are the blends shown in Fig. 1 for *wonder woman* vs *woman wonder* and *apple man* vs *man apple*. Despite already having this in consideration, we think that this connection between name and representation deserves to be further developed, in order to better understand what makes the blend better represent the concept (Pollak et al. 2015).

Results are not always easy to understand. An example of this are the results obtained when introducing the concept *dog* (see Fig. 6). To propose blends for the initial concept, the system makes connections to other concepts. In the case of *dog*, the related concepts are: *drink water*, *guard house* and *sense danger*. Even though all these make sense for describing *dog*, it is not easy to perceive *dog* just by looking at them.

Current issues The blends produced do not always make sense and cannot be considered good representations for the introduced concept– e.g. in Fig. 6 the concept *unicorn* is extended to *spiral horn*, which then leads to a shell emoji (for *spiral*) and a postal horn emoji (for *horn*). In other cases, the search for related concepts even leads to opposite meanings. This results in the generation of blends that do not represent the introduced concept but something that represents its opposite instead. One example of this is the blend for the concept *cold*, in which a candle is represented (see Fig. 6). Additionally, not all aspects are considered. For example, plurals do not affect the blends in most cases and the removal of stop-words affects the meaning (e.g. *Serpent of the Year* is not the same as *Serpent Year* but the system considers them as equal). These issues make it necessary to further improve the approach in terms of linguistic analysis. However, that was not the focus of this paper and, as such, we do not see these issues as problematic but as future work.

Comparison with previous work

This project can be considered a development of our previous work (Cunha et al. 2017) in the way that both deal with Visual Blending. One major advantage of our approach is that it has a very wide conceptual reach (depending only on the emoji knowledge), whereas in Cunha et. al (2017) the system was limited to the concepts *pig*, *angel* and *cactus*. On the other hand, the present work does not involve Conceptual Blending. We plan on adding semantic information to the initial emoji images, allowing us to implement conceptual blending and thus change the system into a Visual Conceptual Blender.

In comparison to previous research on emoji generation, in which the results distorted by visual noise, we were able to obtain blends of high quality, similar to existing emoji.

Conclusion and future work

We propose a system which has the main goal of generating new emoji by using Visual Blending and Semantic Network exploration. Current state of the art was described, focusing on Emoji and Visual Blending. The architecture of the system was presented and the different system components were explained. In order to assess the quality of the blend generation process, a user study was conducted, which focused on three things: ability to represent concepts, quality of the blends and degree of surprise. Overall the system was able to produce concept-representative emoji and, for many cases, the participants stated that the blends were different from what they were expecting.

Future enhancements to the proposed approach include: (i) increasing the number of words for the concept introduced by the user; (ii) implementing a process of conceptual blending based on Cunha et al. (2017) but also blend evaluation, e.g. (Martins et al. 2015); (iii) defining a fitness function for automatic assessment of blend quality and possibly implementing guided evolution; (iv) cleaning the semantic knowledge from EmojiNet, specially the emoji descriptions which have unuseful information; and (v) exploring blend naming, e.g. (Pollak et al. 2015).

Link The system described in this paper is used on the platform *Emojinating*, which will be available at `http://rebrand.ly/emojinating`.

Acknowledgments

João M. Cunha is partially funded by Fundação para a Ciência e Tecnologia (FCT), Portugal, under the grant SFRH/BD/120905/2016.

This work includes data from ConceptNet 5, which was compiled by the Commonsense Computing Initiative. ConceptNet 5 is freely available under the Creative Commons Attribution-ShareAlike license (CC BY SA 4.0) from http://conceptnet.io. The included data was created by contributors to Commonsense Computing projects, contributors to Wikimedia projects, Games with a Purpose, Princeton University's WordNet, DBPedia, OpenCyc, and Umbel.

References

Abbing, R. R.; Pierrot, P.; and Snelting, F. 2017. Modifying the universal. *Executing Practices* 33.

Ai, W.; Lu, X.; Liu, X.; Wang, N.; Huang, G.; and Mei, Q. 2017. Untangling emoji popularity through semantic embeddings. In *ICWSM*, 2–11.

Barbieri, F.; Espinosa-Anke, L.; Ballesteros, M.; Saggion, H.; et al. 2017. Towards the understanding of gaming audiences by modeling twitch emotes. In *Third Workshop on Noisy User-generated Text (W-NUT 2017); 2017 Sep 7; Copenhagen, Denmark. Stroudsburg (PA): ACL; 2017. p. 11-20.* ACL (Association for Computational Linguistics).

Barbieri, F.; Ronzano, F.; and Saggion, H. 2016. What does this emoji mean? a vector space skip-gram model for twitter emojis. In *LREC*.

Berov, L., and Kuhnberger, K.-U. 2016. Visual hallucination for computational creation. In *Proceedings of the Seventh International Conference on Computational Creativity*.

Confalonieri, R.; Corneli, J.; Pease, A.; Plaza, E.; and Schorlemmer, M. 2015. Using argumentation to evaluate concept blends in combinatorial creativity. In *Proceedings of the Sixth International Conference on Computational Creativity*, 174–181.

Correia, J.; Martins, T.; Martins, P.; and Machado, P. 2016. X-faces: The exploit is out there. In *Proceedings of the Seventh International Conference on Computational Creativity*.

Cunha, J. M.; Gonçalves, J.; Martins, P.; Machado, P.; and Cardoso, A. 2017. A pig, an angel and a cactus walk into a blender: A descriptive approach to visual blending. In *Proceedings of the Eighth International Conference on Computational Creativity (ICCC 2017).*

Danesi, M. 2017. *The semiotics of emoji: The rise of visual language in the age of the internet*. Bloomsbury Publishing.

Dimson, T. 2015. Emojineering part 1: Machine learning for emoji trends.

Donato, G., and Paggio, P. 2017. Investigating redundancy in emoji use: Study on a twitter based corpus. In *Proceedings of the 8th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, 118–126.

Dürscheid, C., and Siever, C. M. 2017. Beyond the alphabet–communcataion of emojis. *Kurzfassung eines (auf Deutsch) zur Publikation eingereichten Manuskripts*.

Eisner, B.; Rocktäschel, T.; Augenstein, I.; Bosnjak, M.; and Riedel, S. 2016. emoji2vec: Learning emoji representations from their description. In *Proceedings of The Fourth International Workshop on Natural Language Processing for Social Media*, 48–54.

Fauconnier, G., and Turner, M. 2002. *The Way We Think*. New York: Basic Books.

Gatys, L. A.; Ecker, A. S.; and Bethge, M. 2015. A neural algorithm of artistic style. *arXiv preprint arXiv:1508.06576*.

Goguen, J. 1999. An introduction to algebraic semiotics, with applications to user interface design. In *Lecture Notes in Artificial Intelligence*, volume Computation for Metaphor, Analogy and Agents, 242–291. Springer.

Guibon, G.; Ochs, M.; and Bellot, P. 2016. From emojis to sentiment analysis. In *WACAI 2016*.

Gustafsson, V. 2017. Replacing words with emojis and its effect on reading time. *USCCS 2017* 73.

Heath, D., and Ventura, D. 2016. Before a computer can draw, it must first learn to see. In *Proceedings of the 7th International Conference on Computational Creativity, page to appear*.

Hu, T.; Guo, H.; Sun, H.; Nguyen, T.-v. T.; and Luo, J. 2017. Spice up your chat: The intentions and sentiment effects of using emoji. *arXiv preprint arXiv:1703.02860*.

Lebduska, L. 2014. Emoji, emoji, what for art thou?

Liapis, A. 2018. Recomposing the pokémon color palette. In *Applications of Evolutionary Computation*. Springer.

Martins, P.; Urbancic, T.; Pollak, S.; Lavrac, N.; and Cardoso, A. 2015. The good, the bad, and the aha! blends. In *6th International Conference on Computational Creativity, ICCC 2015*.

McCaig, G.; DiPaola, S.; and Gabora, L. 2016. Deep convolutional networks as models of generalization and blending within visual creativity. In *Proceedings of the Seventh International Conference on Computational Creativity*.

Miller, H.; Thebault-Spieker, J.; Chang, S.; Johnson, I.; Terveen, L.; and Hecht, B. 2016. Blissfully happy or ready to fight: Varying interpretations of emoji. *Proceedings of ICWSM* 2016.

Navigli, R., and Ponzetto, S. P. 2012. Babelnet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence* 193:217–250.

Niediek, I. 2016. Don't write it, picture it!: Accessible information by graphic signs. In *Proceedings of the 7th International Conference on Software Development and Technologies for Enhancing Accessibility and Fighting Info-exclusion*, 188–193. ACM.

Novak, P. K.; Smailović, J.; Sluban, B.; and Mozetič, I. 2015. Sentiment of emojis. *PloS one* 10(12):e0144296.

Pereira, F. C., and Cardoso, A. 2002. The boat-house visual blending experience. In *Proceedings of the Symposium for Creativity in Arts and Science of AISB 2002*.

Pereira, F. C. 2007. *Creativity and Artificial Intelligence: A Conceptual Blending Approach*. Berlin: Mouton de Gruyter.

Phillips, B. J., and McQuarrie, E. F. 2004. Beyond visual metaphor: A new typology of visual rhetoric in advertising. *Marketing theory* 4(1-2):113–136.

Pohl, H.; Domin, C.; and Rohs, M. 2017. Beyond just text: Semantic emoji similarity modeling to support expressive communication. *ACM Transactions on Computer-Human Interaction (TOCHI)* 24(1):6.

Pollak, S.; Martins, P.; Cardoso, A.; and Urbancic, T. 2015. Automated blend naming based on human creativity examples. In *23rd International Conference on Case-Based Reasoning (ICCBR 2015) Workshop on Experience and Creativity*.

Puyat, M. 2017. Emotigan: Emoji art using generative adversarial networks. CS229: Machine Learning Course, Stanford University.

Radpour, D., and Bheda, V. 2017. Conditional generative adversarial networks for emoji synthesis with word embedding manipulation. *arXiv preprint arXiv:1712.04421*.

Ribeiro, P.; Pereira, F. C.; Marques, B.; Leitao, B.; and Cardoso, A. 2003. A model for creativity in creature generation. In *4th International Conference on Intelligent Games and Simulation (GAME-ON 2003)*.

Rodrigues, D.; Prada, M.; Gaspar, R.; Garrido, M. V.; and Lopes, D. 2018. Lisbon emoji and emoticon database (leed): norms for emoji and emoticons in seven evaluative dimensions. *Behavior research methods* 50(1):392–405.

Speer, R., and Havasi, C. 2012. Representing general relational knowledge in conceptnet 5. In *LREC*, 3679–3686.

Steinbrück, A. 2013. *Conceptual blending for the visual domain*. Ph.D. Dissertation, Masters thesis, University of Amsterdam.

Taigman, Y.; Polyak, A.; and Wolf, L. 2016. Unsupervised cross-domain image generation. *arXiv preprint arXiv:1611.02200*.

Wicke, P. 2017. Ideograms as semantic primes: Emoji in computational linguistic creativity.

Wijeratne, S.; Balasuriya, L.; Sheth, A.; and Doran, D. 2017a. Emojinet: An open service and api for emoji sense discovery. In *11th International AAAI Conference on Web and Social Media (ICWSM)*, 437–446.

Wijeratne, S.; Balasuriya, L.; Sheth, A. P.; and Doran, D. 2017b. A semantics-based measure of emoji similarity. In *Proceedings of the International Conference on Web Intelligence, Leipzig, Germany, August 23-26, 2017*, 646–653.

Xiao, P., and Linkola, S. 2015. Vismantic: Meaning-making with images. In *Proceedings of the 6th Int. Conference on Computational Creativity, ICCC-15*.

Explainability: An Aesthetic for Aesthetics in Computational Creative Systems

Paul M. Bodily and Dan Ventura
Computer Science Department
Brigham Young University
Provo, UT 84602 USA
paulmbodily@cs.byu.edu,ventura@cs.byu.edu

Abstract

Of continued interest in the field of Computational Creativity (CC) is the question of what characteristics are required for autonomous creativity. Many characteristics have been proposed including the possession of an autonomous aesthetic. Paramount to the idea of an autonomous aesthetic is the need for a meta-aesthetic: an aesthetic which guides the system in selecting its own aesthetic. We review how aesthetics have (and have not) been used in CC systems to date, including examples of autonomous aesthetics. We formalize the idea of a meta-aesthetic in an extension of Wiggins' 2006 framework for describing computational systems generally. We propose *explainability* as an effective meta-aesthetic for autonomous creative systems and make some comments about the explainability of creativity and of explainability itself.

Introduction

Computational creativity (CC) has been characterized as the quest for "computational systems which, by taking on particular responsibilities, exhibit behaviours that unbiased observers would deem to be creative" (Colton and Wiggins 2012). Since the dawn of CC, researchers have continually hypothesized about what, in addition to output alone, is required to create the perception of creativity in a computational system, suggesting such elements as creative processes (Ritchie 2007; Colton 2008), self-evaluation (Wiggins 2006; Jennings 2010), intention (Ventura 2016; Jordanous and Keller 2016; Guckelsberger, Salge, and Colton 2017), and self-awareness (Linkola et al. 2017). These aspects of a creative system, which often imitate characteristics of human creativity, lead the observer to sense that the system has a consciousness which impels its creative behavior.

The high goal of conscious creativity hinges to a large extent on the ability of a system to create its own *aesthetic*. Deriving from the Greek word *aisthetikos* meaning "sentient" or "perceptive", this term was coined in Alexander Baumgarten's 1735 dissertation *Meditationes philosophicae de nonnullis ad poema pertinentibus* (Philosophical considerations of some matters pertaining to the poem) to describe art as a means of knowing. An aesthetic describes a philosophy of art, a theory of criticism, a set of values or beliefs about what is beautiful and good (Mothersill 2004).

Among the definitions for *aesthetic* listed by Koren in his book *Which "aesthetics" do you mean?: Ten definitions* we are interested in those that define an aesthetic as: a "cognitive mode" or awareness of abstract and/or particular "sensory and emotive qualities"; an "opinion, belief, or attitude related to some of the underlying principles of art" which, if not explicit, can be inferred from artifacts; a "style" or "perceptually cohesive organization of qualities...that is distinct from other perceptually cohesive organizations of qualities"; and an "ability to make judgments of value" (2010). In short, we think of an aesthetic as an opinion, belief, or attitude about principles of art (in the broadest sense of the term) of which there is some cognitive awareness and which serves to make judgments of value related to style.

Some have attempted to distinguish between the evaluation of creativity (with emphasis on novelty) and aesthetics (with emphasis on pleasure or beauty) (Cohen et al. 2012). This distinction is valid when *aesthetics* is used to refer to "superficial appearance" or as a synonym for *taste* or *beauty* (Koren 2010); however, in the definition we adopt an *aesthetic* encompasses all qualities that are necessary to judge a piece of art as successful or, in our case, creative.

It has also been noted that in considering aesthetics as an "evaluative discipline," there is a distinction between judgment and evaluation (Cohen et al. 2012): whereas *evaluations* represent technical, objective, quantifiable *measures* (e.g., 37° C), *judgments* represent "human", subjective, qualitative *values* (e.g., "it's hot"). This distinction poses a critical challenge for discussing aesthetics in computational systems (which by nature lack human subjectivity) and raises questions about whether qualitative judgments merely represent or can somehow be represented by complex quantitative evaluations. This discussion is beyond our scope, but for the purposes of this paper we assume that an aesthetic judgment can be represented as a quantitative function.

The idea of incorporating an aesthetic into a computational system has been broadly discussed and many CC systems (implicitly or explicitly) define aesthetics. So also has the topic of initiating and changing a system's aesthetic been addressed in various places (e.g., (Jennings 2010)). While many have advocated the enhanced creativity of a system which possesses its own aesthetic, we find that little has been said about what principles can be used to guide the system in selecting of a "good" aesthetic. In short what is needed is an aesthetic for aesthetics.

In what follows we review what has been said elsewhere about aesthetics and autonomous aesthetics as they relate to computational systems. We also review what aesthetics have been proposed for CC systems. We finally turn to the idea of a *meta-aesthetic*, or an aesthetic for evaluating aesthetics, and propose *explainability* as one such meta-aesthetic. Our purpose is two-fold: first, to bring attention and add fuel to the assertion that a system that is aesthetically autonomous is more creative than one that is not; and second, to argue that because creativity is a fundamentally social construct, explainability is a critical characteristic of a creative system's meta-aesthetic.

Background: Aesthetics in CC Systems

The concept of an aesthetic (as we have defined it above) has been referenced using a variety of terms in the CC literature. Papadopoulos and Wiggins lament that "the big disadvantage of most, if not all, the computational models (in varying degrees) is that... the computers do not have *feelings*" (emphasis added) (1999). Boden, in her seminal work on creativity and artificial intelligence, talks of a "pre-existing mental structure" or " hidden mental faculty which has positive evaluation built in to it" (2004). In describing a framework of crucial properties for creative systems, Wiggins concretizes this built-in "positive evaluation" as a "*set of rules*" $\mathcal{E}$ for evaluating concepts "according to whatever criteria we may consider appropriate" (2006). Jennings describes autonomous systems as possessing the ability to initiate and guide changes to its "standards" and generate its own "opinions" (2010). Each of these terms highlight aspects that we have previously identified as defining an aesthetic.

Attempts at formalizing computational aesthetics span nearly a century. George David Birkhoff is credited with having fathered computational aesthetics in his 1933 book *Aesthetic Measure* in which he defines aesthetic measure as the ratio of *order* to *complexity*. From this definition came other aesthetic measurements including Shannons entropy (Shannon 2001) and Kolmogorov complexity (Rigau, Feixas, and Sbert 2007). When applied to concrete aspects of an artefact these measures represent aesthetics evaluations that have been used in many computational systems.

On the assumption that an aesthetic encompasses all qualities that are necessary to judge an artefact as creative, two common aesthetic qualities used in the judgment of CC artefacts are novelty and typicality (Ritchie 2007). One way that these qualities have been measured is the Wundt curve (Berlyne 1970) (see Figure 1). The curve represents the value of an artefact as novelty increases. Initially value increases as new ideas and features are incorporated into the artefact. At some point, however, the artefact becomes so new that it begins to no longer fit within the domain of interest and the value decreases until it is no longer of interest.

In addition to *novelty* and *typicality*, several other aesthetic values have been presented in the CC literature including: *skill*, *imagination*, and *appreciation* (Colton 2008),

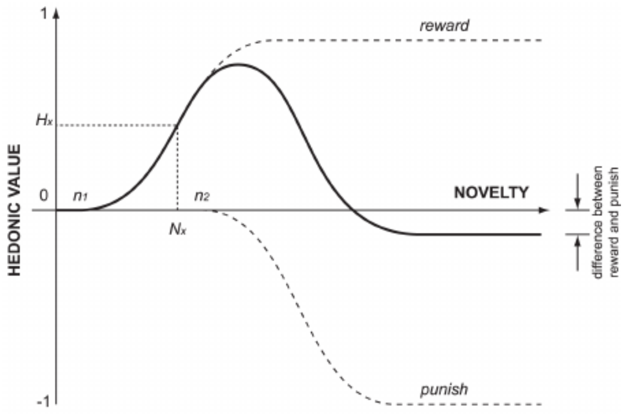


Figure 1: The Wundt curve represents an example of a computational aesthetic quality commonly used for evaluating novelty and typicality. As the novelty of an artefact increases, so *to an extent* does the value. At some point the artefact strays far enough from the bounds of "normal" to prevent observers from recognizing even limited value. Figure from Saunders et al. (2010).

as well as *value* and *surprise* (Boden 2004). Hofstadter suggests that *complexity* is an aesthetic for creativity (1980).

Every creative system is reflective of an aesthetic (Koren 2010), and in some cases the aesthetic is explicitly modeled. CC systems have used a number of different aesthetic implementations including neural networks trained on user ratings (Morris et al. 2012; Monteith, Martinez, and Ventura 2010; Heath and Ventura 2016); Markov models (Abgaz et al. 2017); and "baked-in" knowledge (Ventura 2016) and many others.

It is a common assumption that possessing an autonomous aesthetic, independent from that of the programmer or designer, is a fundamental characteristic of creative systems. Colton (2008) asserts that ultimately the perception of creativity will require that CC systems "develop their own aesthetic along with their own styles". Guckelsberger, Salge, and Colton propose moving away from anthropocentric models and present the *enactive artificial intelligence* framework as a minimal model of intentional creative agency (2017).

Systems can be found that to varying extents develop an autonomous aesthetic. Cook and Colton (2015) present a painting system that evolves its own preference functions which enable it to make non-random, consistent aesthetic choices that are not based on any external, existing opinion. DARCI implements separate semantic and image generation models and uses the semantic model autonomously to guide the process of rendering images that convey particular concepts, including those not seen in training (Heath and Ventura 2016). Jennings (2010) argues that a creative system must not only have an autonomous aesthetic, but must also be capable of autonomously *changing* its own aesthetic in non-random ways (see also Ackerman et al. (2017)).

A Framework for Describing Aesthetics

As several in the field have asserted (and which we also assert), an autonomous aesthetic is a key component of creative systems. This assertion naturally begs the question: What are the characteristics that a computational system might use for *selecting* an aesthetic?

To more precisely consider this question, we present a framework for describing, analyzing, and comparing aesthetics. The framework might be thought of as being analogous and a possible extension to the framework of Wiggins (2006), but with focus on aesthetics instead of concepts. In Wiggins' framework for describing, analyzing, and comparing concepts, a set of rules $\mathscr{E}$ for evaluating concepts is presented as a crucial component of creative systems. The function computed by $\mathscr{E}$—denoted by Wiggins as $[[\mathscr{E}]]$—yields a real numbered value in the range [0,1] representing the distribution of evaluation scores $[[\mathscr{E}]](c)$ over all concepts c in the concept universe. While Wiggins chooses to forgo any discussion on what is included in $\mathscr{E}$, we make the simplifying assumption (similar to (Jennings 2010)) that $[[\mathscr{E}]]$ is equivalent to a system's standards or aesthetic (we later discuss the validity of this assumption). It is also significant to our discussion to note that a language, $\mathscr{L}$, is needed in order to express this set of rules $\mathscr{E}$ (Wiggins 2006).

Let us hypothesize that there exists an aesthetic universe $\mathscr{A}$ which encompasses all possible aesthetics.

Definition 1 (*Aesthetic universe*). The *aesthetic universe*, $\mathscr{A}$, is a multidimensional space, whose dimensions allow for the representation of any aesthetic, and all possible distinct aesthetics correspond with distinct points in $\mathscr{A}$.

A more precise formulation of an aesthetic a is beyond the scope of this paper and when discussing those aesthetics included in $\mathscr{A}$ we include abstract and concrete, partial and complete aesthetics.

Axiom 1 (*Aesthetic universality*). All possible aesthetics are represented in $\mathscr{A}$; thus, $\mathscr{A}$ is the type of all possible aesthetics.

Axiom 2 (*Non-identity of aesthetics*). All aesthetics, a_i, represented in $\mathscr{A}$ are mutually non-identical. $\forall a_1, a_2 \in \mathscr{A}, a_1 \neq a_2$.

We now define a set of rules, $\mathscr{T}_{\mathscr{A}}$, which allows a traversal of $\mathscr{A}$ according to some search strategy and a set of rules, $\mathscr{E}_{\mathscr{A}}$, for evaluating the quality of any aesthetic found. $\mathscr{A}$ can be enumerated using an interpreter $\langle\langle ., . \rangle\rangle$, which, given $\mathscr{T}_{\mathscr{A}}$ and $\mathscr{E}_{\mathscr{A}}$, maps an ordered subset of $\mathscr{A}$, a_{in}, to another ordered subset of $\mathscr{A}$, a_{out}:

$$a_{out} = \langle\langle \mathscr{T}_{\mathscr{A}}, \mathscr{E}_{\mathscr{A}} \rangle\rangle(a_{in}).$$

In essence, $\mathscr{T}_{\mathscr{A}}$ might be thought of as a strategy for mutating a system's aesthetic as a function of previous aesthetics and $\mathscr{E}_{\mathscr{A}}$ as an evaluation mechanism for aesthetics. It is beyond our scope to consider what $\mathscr{T}_{\mathscr{A}}$ might look like, though others have devoted some significant thought to this idea (e.g., Jennings(2010)).

The discussion on which we choose to focus revolves instead around the set of rules for evaluating aesthetics, $\mathscr{E}_{\mathscr{A}}$, and the language, $\mathscr{L}$, which is used to express a particular aesthetic, a_i.

Explainability: An Aesthetic for Aesthetics

What makes a good aesthetic? As evidenced by the variety of aesthetics implemented (implicitly or explicity) in extant CC systems, many have pondered what good aesthetics for CC systems might be. *Our* purpose is rather to consider what all "good" aesthetics have in common.

Our proposal for what makes a good meta-aesthetic hinges on the idea that *creativity is an inherently social construct*. In his book on creativity Csikszentmihalyi writes: "Creativity does not happen inside people's heads, but in the interaction between a person's thoughts and a sociocultural context" (1996). He also comments: "Over and over again, the importance of seeing people, hearing people, exchanging ideas, and getting to know another person's work and mind are stressed by creative individuals."

To push this point further, let us consider for a moment a slightly adapted version of Colton's allegory of the *'Dots 2008'* exhibit (2008). In the original allegory, two painters display paintings composed of a "seemingly random arrangement of dots of paint". The story goes that despite the fact that they appear identical, an observer falls in love with one painting when its artist explains that unlike his colleague, whose painting represents nothing more than randomly placed dots, in *his* painting "each dot represents a friend of mine. The colour of the dot represents how I feel about them, and the position indicates how close I am to them." Colton uses this allegory to illustrate that creativity lies as much in the process as in the output.

To adapt the allegory, consider now that both paintings were inspired to represent the painters' feelings toward their friends. In this version of the story, the art-lover then asks each painter: "What made you decide to paint your feelings towards your friends?" The first painter responds: "My friends are important to me." The second shrugs, gestures towards the first, and responds: "My art teacher told me to." Returning a week later with a friend, the art-lover explains to the friend that both paintings represent the creativity of the first artist.

The original version of the allegory was used by Colton to demonstrate that the perception of creativity depends as much on the explanation of the creative process as it does on the artefact itself. The *extended* version of the allegory emphasizes that the perception of creativity depends equally as much on explanation of the *aesthetic*.

This line of thought leads us to the proposition of *explainability* as an aesthetic for aesthetics, that is, the idea that a good aesthetic can be *explained*. In support of this idea, it is interesting to note that in some contexts the term *aesthetic* is even defined as "the verbal or written statement itself" of beliefs about art (Koren 2010).

To our knowledge very few systems—and no systems with an autonomous aesthetic—exist which attempt to ex-

Figure 2: In its *You Can't Know my Mind* exhibit, The Painting Fool paints portraits reflective of a "mood" and possibly an aesthetic. Figure from Colton and Ventura (2014).

plain their aesthetic, though it is not uncommon to see presuppositions about a "system [that] has unlimited capacities to enter into a dialogue and to frame its actions" (2017). Two notable exceptions to the dearth of systems that explain their aesthetic are the *You Can't Know my Mind* exhibit from The Painting Fool (Colton and Ventura 2014) and the latest version of DARCI's image creation process (Heath and Ventura 2016).

Colton and Ventura's *You Can't Know my Mind* exhibit features The Painting Fool as it paints (or chooses not to paint) portraits that reflect its current mood and aesthetic (see Figure 2). In addition to displaying the painting, it explains some of what the system was "thinking" and "feeling" as it painted the portrait as well as how well it felt (with the help of DARCI) that it accomplished its intention. Though the system does reflect and explain an autonomous mood and interpretation of particular descriptors, this explanation is arguably not representative of an aesthetic because the system is explaining more its observations and logic rather than its system of values.

DARCI (Heath and Ventura 2016) is another example of a system that approaches the threshold of explaining its aesthetic (see Figure 3). The system begins with an inspiring image from which it tries to "think" of similar looking objects. It then creates a new image of a similar looking object that has been stylistically modified to reflect an aesthetic quality similar to the original. Its explanation of this process, like that of The Painting Fool, focuses primarily on describing the logical process of creating the image, but does give an impression of having consciously thought about aesthetic qualities in its creativity.

These two examples suffice to demonstrate that an explainable aesthetic *can* contribute to the perception of a sentient, aesthetically-driven creative system. We await future work to provide corroborative empirical evidence.

Enabling a system to explain its own aesthetic adds a potentially significant degree of effort. But its importance cannot be overstated. Like an overly involved parent, a CC researcher that does not equip a system with the ability to explain its own aesthetic (and is rather constantly butting in to do the explaining themselves) creates a crutch for their systems, never fully realizing the lack of creativity in their offspring until it is left to flounder on its own.

An explainable aesthetic is notably different from an explained aesthetic. Often humans participate in creativity without ever explaining their aesthetics. But this is different than assuming that their aesthetics are not explainable. Many, including Boden (2004), have argued that humans can be creative without being able to explain the aesthetic that motivates their creativity. We would argue, however, that for any human aesthetic *some* degree of explainability exists, even if it be as unconventional as "random" or "anti-aesthetic". We discuss this more below.

The purpose of an aesthetic is to impose *value*. An *explainable* aesthetic makes it possible for a creator to communicate this value to others. An *explained* aesthetic makes it possible for others to understand *why* an artifact is valued and possibly, then, to appreciate it (more) themselves.

A Language for Explaining Autonomous Aesthetics

Explainability of an aesthetic requires a language in which to express the aesthetic. To introduce what that language might be, consider that Wiggins defines a language $\mathscr{L}$ for representing $\mathscr{E}$. How does $\mathscr{E}$ differ from the aesthetic represented by $[[\mathscr{E}]]$? The answer lies in the fact that there could be many rule sets $\mathscr{E}_i$ which describe the *same* function (i.e., domain, co-domain, and range are equal). Considering that each of these rule sets is explained using a language $\mathscr{L}$, this essentially means that there could be many explanations of an aesthetic that are functionally equivalent. How then does a CC system decide which explanation is best? Given two distinct rule sets $\mathscr{E}_1$ and $\mathscr{E}_2$, let $\mathscr{E}_1 \sim \mathscr{E}_2$ mean that $[[\mathscr{E}_1]] = [[\mathscr{E}_2]]$. If $\mathscr{E}_1 \neq \mathscr{E}_2$ but $\mathscr{E}_1 \sim \mathscr{E}_2$, which explanation is to be preferred?

One likely suggestion is to consider the relationship between the amount of information contained within the explanation and the length of the explanation. The Kolmogorov complexity K, which is related to Shannon's entropy H but is *language-aware*,[1] provides just such a measure and has

[1]Note that there is an important subtlety here. There is a description language $\mathscr{D}$ associated with the definition of K. This should *not* be confused with the explanation language $\mathscr{L}$.

(a) source (b) example (c) sketch (d) rendered (e) **Overdress**

Figure 3: An example showing the intermediate images of each step of DARCI's image creation process. A generated description (personifying the system) is as follows: "I was looking for inspiration from this image (a), And it made me feel **gloomy** and **dreamy**. It also made me think of this image that I've previously seen (b), which is a picture of a **poncho**. So I started an initial image of my own by searching for a background image on the Internet based on **poncho**, **gloomy**, and **dreamy**. Then I took basic iconic images associated with those concepts and resized/placed them on the background according to how relevant they were. This was the result (c). I then modified it in a style related to **poncho**, **gloomy**, and **dreamy**, which resulted in this image (d). I did a final modification based on aesthetic quality and how closely the style related to the original image (e). The end result perhaps looks more like a **cloak** or a **vestment**, and it feels particularly **gloomy**. I call it **Overdress**."

been suggested in this capacity before (Rigau, Feixas, and Sbert 2007). The Kolmogorov complexity $K(\mathscr{E})$ of an explanation $\mathscr{E}$ of an aesthetic is distinct from the complexity of the aesthetic $[[\mathscr{E}]]$ itself (which might be measured using entropy H). Returning to the equivalent explanations $\mathscr{E}_1, \mathscr{E}_2$, this means in general that because $\mathscr{E}_1 \sim \mathscr{E}_2$, $H(\mathscr{E}_1) = H(\mathscr{E}_2)$, but because $\mathscr{E}_1 \neq \mathscr{E}_2$, $K(\mathscr{E}_1) \neq (\mathscr{E}_2)$, and the explanation with lower complexity is to be preferred.

K and H, however, are related through $\mathscr{L}$—while H is language-agnostic, K is not. If $\mathscr{E}_1$ and $\mathscr{E}_2$ are expressed in the same language, this is perhaps not noteworthy; however, if they are expressed in distinct languages $\mathscr{L}_1$ and $\mathscr{L}_2$, things become more interesting. If we think of the set of all possible aesthetics $\mathscr{A}$ (i.e., the set of all possible $[[\mathscr{E}]]$ or the set of all possible distributions over the concept universe), then depending on the choice of $\mathscr{L}$, aesthetics of high or low entropy may be considered good. In other words, the complexity of the aesthetic $H([[\mathscr{E}]])$ is not correlated with its goodness; rather, the complexity of its *explanation* $K(\mathscr{E})$ is, and that is language dependent.

This suggests at least two other ideas:

- The invention of new languages $\mathscr{L}$ is as interesting as the invention of new aesthetics $[[\mathscr{E}]]$ (as touched on by (Saunders and Grace 2008)).
- The discovery of new $\mathscr{E}$ is not the same as the discovery of new $[[\mathscr{E}]]$. It is still useful though, as reductions in $K(\mathscr{E})$ signify improved communicability.

The idea that particular languages encourage particular aesthetics is well-known in the realm of natural language. It has long been known that language guides cognition and that conceptual knowledge is shaped by a person's language (Whorf et al. 2012). It has also been shown, for example, that people make different choices based on whether the decision is framed in a native or foreign language (Keysar, Hayakawa, and An 2012). Words with significant cultural and aesthetic value often have no ("good") translation from one language to another. Honorifics which exist in some languages are absent in others. Even grammars themselves can be indicative of a culture's belief system.

The variable impact of language on the goodness or complexity of an explanation leads us to postulate a "no free lunch theorem of aesthetic languages" (which we do not attempt to prove).

Theorem 1 *The no free lunch theorem of aesthetic languages.* Given two languages, $\mathscr{L}_1$ and $\mathscr{L}_2$,

$$\sum_{[[\mathscr{E}]] \in \mathscr{A}} K(\mathscr{E}^*_{\mathscr{L}_1}) = \sum_{[[\mathscr{E}]] \in \mathscr{A}} K(\mathscr{E}^*_{\mathscr{L}_2})$$

where $\mathscr{E}^*_{\mathscr{L}_1}$ and $\mathscr{E}^*_{\mathscr{L}_2}$ are explanations of aesthetic $[[\mathscr{E}]]$ in languages $\mathscr{L}_1$ and $\mathscr{L}_2$, respectively, and $K(\mathscr{E}^*_{\mathscr{L}}) \leq K(\mathscr{E}'_{\mathscr{L}})$ for all $\mathscr{E}'_{\mathscr{L}} \sim \mathscr{E}^*_{\mathscr{L}}$. That is, $\mathscr{E}^*_{\mathscr{L}_1}$ ($\mathscr{E}^*_{\mathscr{L}_2}$) is the least complex explanation for aesthetic $[[\mathscr{E}]]$ in language $\mathscr{L}_1$ ($\mathscr{L}_2$).

In other words, if a language admits low complexity of explanation for some set of aesthetics $\mathscr{B} \subset \mathscr{A}$ then it necessarily pays for that with unavoidably higher complexity explanations for the set of all remaining aesthetics $\mathscr{A} \backslash \mathscr{B}$.

This leads to another important point about explainability: to the extent that the *shared* language between a system and its audience is different, its ability to share aesthetics will be limited (cf. (Saunders and Grace 2008)).

Does Wiggins' $[[\mathscr{E}]]$ compute an Aesthetic?

We have previously assumed that the function $[[\mathscr{E}]]$ computed by interpreting the rule set $\mathscr{E}$ is equivalent to some aesthetic $a \in \mathscr{A}$. We now consider the argument that the function $[[\mathscr{E}]]$ is instead somehow distinct from any aesthetic. To prove this argument we need to demonstrate an instance in which $[[\mathscr{E}]]$ differs from that of the aesthetic a. This would suggest that there are other factors besides a which determine $[[\mathcal{E}]]$. We believe that most scenarios which might demonstrate this contradiction fit into one of two categories, which we call *domain rectification* and *aesthetic transfer*. Prior to summarizing the general characteristics of these categories, we share one thought experiment representative of each category.

History and Allegory Imagine a scholar who encounters a volume of ancient text. The scholar, who values historical accuracy, determines in the course of reading the text that the events described could not be historically accurate and dismisses the text as being of questionable value. A short time later, the scholar is informed by a friend that the volume in question was intended as an allegory rather than as an historical account. Rereading the text through this lens, she now finds value in the insights afforded by the allegorical interpretation.

It might appear in this example that the scholar's evaluation, $[[\mathcal{E}]](c)$, of the text c changed while her aesthetic, a, did not. We would argue, however, that there was no change in either $[[\mathcal{E}]](c)$ or a. Rather the text is being recategorized into a different domain resulting in a different evaluation $[[\mathcal{E}]]'$. In defining his framework, Wiggins' notably states that $\mathcal{E}$ is used to evaluate concepts *within a specific concept domain*, not the greater concept universe (2006). Seen in this light, there are in fact two different evaluation functions ($[[\mathcal{E}_{history}]]$ and $[[\mathcal{E}_{allegory}]]$) and also two different aesthetics ($a_{history}$ and $a_{allegory}$) at play here. The parallel changes from $[[\mathcal{E}_{history}]]$ to $[[\mathcal{E}_{allegory}]]$ and from $a_{history}$ to $a_{allegory}$ provide a plausible explanation that avoids the conclusion that $[[\mathcal{E}]]$ must be distinct from a.

The Moody Young Pianist Imagine a moody young pianist enrolled to study piano. Despite his interest in other forms of music, his instructor insists on teaching him classical music and assigns him to learn Brahms Rhapsody in G Minor Op. 79, No. 2. The boy at first does not like the piece, but in the course of time his instructor invites him to think of the piece as an interpretation through dynamics, tempo, and melodic expression of his own life experiences. This idea excites the boy, who develops a love and ownership for the piece he once despised.

It may seem that although the boy's evaluation $[[\mathcal{E}]](c)$ of the piece c had changed, his aesthetic a did not. Just as $[[\mathcal{E}]]$ is relative to a particular domain, so too (we will assume now and question later) is a domain-dependent. Here, we suggest that an aesthetic a_1 from one domain (e.g., self-expression) is transferred to another domain (e.g., classical music) with had been associated with aesthetic a_2; effectively, a_1 (temporarily or partially) replaces a_2. Thus the apparent change in $[[\mathcal{E}]]$ is associated with a change in aesthetic. This scenario is also plausibly explained without having to conclude that $[[\mathcal{E}]]$ is in any way distinct from a.

These thought experiments are representative of scenarios in which it may appear that the function $[[\mathcal{E}]]$ changes while the aesthetic a does not; however plausible arguments can be made in both cases that avoid this conclusion:

- **Domain rectification:** Apparent changes in $[[\mathcal{E}]]$ actually result from the application of different domain-specific evaluation functions (e.g., $[[\mathcal{E}_i]]$ is replaced by $[[\mathcal{E}_j]]$, where $\mathcal{E}_i \nsim \mathcal{E}_j$).
- **Aesthetic transfer:** Changes in $[[\mathcal{E}]]$ within a concept domain $\mathcal{C}$ occur in association with the (temporary or partial) adoption of an aesthetic a_j usually associated with some other concept domain $\mathcal{C}'$ and its being used in place of a_i for $\mathcal{C}$.

While these thought experiments do not prove our assumption of the equivalence of $[[\mathcal{E}]]$ and a, they do provide some suggestive support for the idea that $[[\mathcal{E}]] = a$. For now we will leave this an open question.

These two scenarios also serve to strengthen the argument for explainability as a meta-aesthetic. In the first scenario, the difference between the evaluations is explained by a change in the *domain* of the aesthetics. Without an explanation of the different domain-specific aesthetics (e.g., "what makes this a good historical account" versus "what makes this a good allegory"), differences in creative evaluation due to subjective opinion (which sometimes causes differences) cannot be distinguished from differences due to contrary assumptions about the contextual domain.

In the second scenario, the difference between the evaluations is explained by a *change* or expansion in the aesthetic. Without an explanation of the differential aesthetics, differences in creative evaluation due to a reversal of subjective opinion (e.g., "Brahms instinctively sounds good to me now") cannot be distinguished from those due to an expanded aesthetic ("I don't typically like Brahms, but this song has added meaning to me").

In both scenarios an explainable aesthetic is needed to allow the system to convincingly demonstrate that autonomous changes to its aesthetic are occurring in nonrandom ways.

The Explainability of Explainability

In proposing an aesthetic for aesthetics an interesting question arises: what happens when you evaluate the meta-aesthetic according to the meta-aesthetic? In this case the question takes on the more concrete form of how well does the aesthetic of explainability hold up under the aesthetic of explainability? How explainable is explainability?

On the one hand the concept of explainability seems readily explicable: ideas and aesthetics that can be communicated are preferred to those which can not. But when push comes to shove, there is a significant double-standard in explainability: though we ask for explainability, we rarely intend for ideas to be explained beyond a few layers of complexity. Indeed the entire discipline of epistemology exists essentially to question the explainability of explainability. Therefore, proposing explainability as a suitable meta-aesthetic is the beginning of a much larger discussion that needs to take place about what degree of explainability is conducive to a discussion of creativity.

Explainability is an interesting topic in relation to *creativity* which is so characteristically unexplainable. In fact many human observers feel that *un*explainability is a critical element of creativity: if the complete process by which an artefact is created is known, then it cannot possibly be creative. Others have argued that creativity emerges when the process extends beyond some sufficient and necessary threshold of complexity (Hofstadter 1980). This is fundamentally at the heart of the debate over mere generation (Ventura 2016). In both humans and computers, too little information will not satiate the observer's desire to understand. On the other hand, too much detail can lead to tedium or (particularly in computers) an impression that the agent is purely carrying out predefined instructions (Colton 2008). Finding that balance is as much a key to the perception of creativity as it is to the discussion of explainability in general.

Discussion and Conclusion

Relevant to the discussion of explaining aesthetics in a given (possibly natural) language, it has been argued that the concept of creativity, though human-conceived, should not remain human-centric (Guckelsberger, Salge, and Colton 2017). While this may be true, it is also true that creativity does not happen in a vacuum, but emerges in the interaction between an agent's "thoughts" and a sociocultural context (Csikszentmihalyi 1996; Bown 2015; Jordanous 2015). It may not be that creativity is human-centric, but until substantial non-human sociocultural contexts are presented, it seems reasonable to expect that creativity in computational systems depends on at least an interaction with human sociocultural contexts.

Certainly, there exist many domains that will always be human-centric, yet to which it would be desirable to have CC agents contributing (e.g., medicine, drug design, autonomous vehicles, etc.) And, it is certain that in many such domains, an ability to explain process and/or product will be demanded by human "consumers".

However, even in a future in which creators and their sociocultural context are wholly non-human, we argue that the notion of explainability would remain a critical consideration, albeit possibly employing non-natural language for that explanation (e.g., see (Saunders and Grace 2008)).

We seem, to a large extent, to have focused thus far as a field on systems which (possibly by some arguably creative process) generate *artefacts*. We do so to the detriment of the field whose stated focus is on the greater umbrella of *behaviors* (Colton and Wiggins 2012). We may find benefit in increasingly promoting research toward systems which, independent of their generative abilities, are creative by virtue of their ability to interact with and internally react to their sociocultural context (see Figure 4).

An interesting open question is whether creativity is domain dependent or whether there is some abstract, core creative mechanism that is domain agnostic. The question is beyond the scope of our current treatment, but it is intimately coupled with the question of whether an aesthetic may be developed and explained independent of a particular creative domain (to which it may eventually be applied).

Systems with an ability to autonomously initiate, change, and explain an (domain-agnostic) aesthetic deriving from a sociocultural context would be a significant contribution to the field of CC, even without (domain-specific) generative capabilities. Such systems, created independently from any particular creative domain, could modularly apply themselves to various domains with compelling intentional creativity and self-evaluation. Such systems could prove useful to those who prefer to focus on more generative aspects of computational creativity by providing an out-of-the-box aesthetic model from which to derive autonomous guidance. Consider, for example, a model-view-controller system (Krasner, Pope, and others 1988) for computational creativity with an abstract model embodying an autonomously initiated and changing aesthetic which is reapplied across multiple domains through domain-specific controllers.

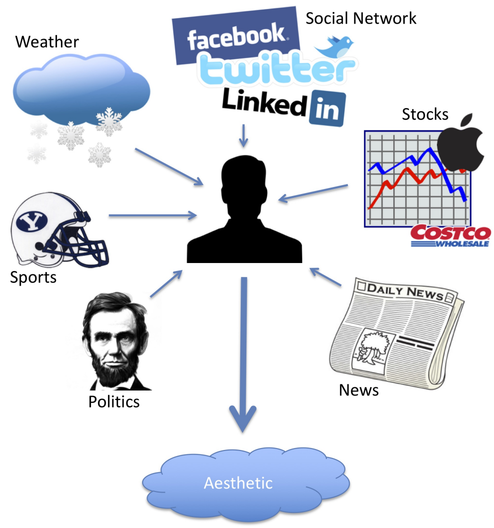


Figure 4: The ability to interact with and internally react to sociocultural context is a fundamental characteristic of creative agents. In addition to interaction with other agents, a typical human sociocultural context might include politics, sports, weather, social media, the economy, and news.

To conclude, we restate the argument of our paper using the readily available analogy of the peer review process by which this paper has been evaluated. Our manuscript is an example of an aesthetic (i.e., our opinions, beliefs, etc.) being explained by creative agents (i.e., the authors) in the demonstration of a creative artefact (i.e., the idea of explainability as a meta-aesthetic)[2]. This example itself demonstrates the thesis of our argument: *aesthetic explainability* is a minimal yet valuable standard to which we hold one another in our own creative endeavors. The success of CC systems will improve as they demonstrate similar capabilities in their own attempts to demonstrate creative behaviors.

References

Abgaz, Y.; Chaudhry, E.; ODonoghue, D.; Hurley, D.; and Zhang, J. J. 2017. Characteristics of pro-c analogies and blends between research publications. In *Proceedings of the 8th International Conference on Computational Creativity*, 1–8.

[2]This qualifies the manuscript as a meta-meta-aesthetic.

Ackerman, M.; Goel, A.; Johnson, C. G.; Jordanous, A.; León, C.; y Pérez, R. P.; Toivonen, H.; and Ventura, D. 2017. Teaching computational creativity. In *Proceedings of the 8th International Conference on Computational Creativity*, 9–16.

Berlyne, D. E. 1970. Novelty, complexity, and hedonic value. *Perception & Psychophysics* 8(5):279–286.

Birkhoff, G. D. 1933. *Aesthetic Measure*. Harvard University Press Cambridge.

Boden, M. A. 2004. *The Creative Mind: Myths and Mechanisms*. Psychology Press.

Bown, O. 2015. Attributing creative agency: Are we doing it right? In *Proceedings of the 6th International Conference on Computational Creativity*, 17–22.

Cohen, H.; Nake, F.; Brown, D. C.; Brown, P.; Galanter, P.; McCormack, J.; and dInverno, M. 2012. Evaluation of creative aesthetics. In *Computers and creativity*. Springer. 95–111.

Colton, S., and Ventura, D. 2014. You can't know my mind: A festival of computational creativity. In *Proceedings of the 5th International Conference on Computational Creativity*, 351–354.

Colton, S., and Wiggins, G. A. 2012. Computational creativity: The final frontier? In *Proceedings of the European Conference on Artificial Intelligence*, volume 12, 21–26.

Colton, S. 2008. Creativity versus the perception of creativity in computational systems. In *AAAI spring symposium: creative intelligent systems*.

Cook, M., and Colton, S. 2015. Generating code for expressing simple preferences: Moving on from hardcoding and randomness. In *Proceedings of the 6th International Conference on Computational Creativity*, 8–16.

Csikszentmihalyi, M. 1996. *Creativity: Flow and the Psychology of Discovery and Invention*. Harper Perennial, New York.

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*, 128–135.

Heath, D., and Ventura, D. 2016. Creating images by learning image semantics using vector space models. In *Proceedings of the Association for the Advancement of Artificial Intelligence*, 1202–1208.

Hofstadter, D. R. 1980. *Gödel, Escher, Bach*. Vintage Books New York.

Jennings, K. E. 2010. Developing creativity: Artificial barriers in artificial intelligence. *Minds and Machines* 20(4):489–501.

Jordanous, A., and Keller, B. 2016. Modelling creativity: Identifying key components through a corpus-based approach. *PLOS One* 11(10):e0162959.

Jordanous, A. 2015. Four pppperspectives on computational creativity. In *Proceedings of the AISB Symposium on Computational Creativity*, 16–22.

Keysar, B.; Hayakawa, S. L.; and An, S. G. 2012. The foreign-language effect: Thinking in a foreign tongue reduces decision biases. *Psychological Science* 23(6):661–668.

Koren, L. 2010. *Which" aesthetics" do you mean?: Ten definitions*. Imperfect Pub.

Krasner, G. E.; Pope, S. T.; et al. 1988. A description of the model-view-controller user interface paradigm in the smalltalk-80 system. *Journal of Object Oriented programming* 1(3):26–49.

Linkola, S.; Kantosalo, A.; Männistö, T.; and Toivonen, H. 2017. Aspects of self-awareness: An anatomy of metacreative systems. In *Proceedings of the 8th International Conference on Computational Creativity*, 189–196.

Monteith, K.; Martinez, T. R.; and Ventura, D. 2010. Automatic generation of music for inducing emotive response. In *Proceedings of the 1st International Conference on Computational Creativity*, 140–149.

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 *Proceedings of the 3rd International Conference on Computational Creativity*, 119–125.

Mothersill, M. 2004. Beauty and the critic's judgment: Remapping aesthetics. In Kivy, P., ed., *The Blackwell Guide to Aesthetics*. Blackwell Publishing Ltd. 152–166.

Papadopoulos, G., and Wiggins, G. 1999. AI methods for algorithmic composition: A survey, a critical view and future prospects. In *Proceedings of the AISB Symposium on Musical Creativity*, 110–117.

Rigau, J.; Feixas, M.; and Sbert, M. 2007. Conceptualizing Birkhoff's aesthetic measure using Shannon entropy and Kolmogorov complexity. In *Proceedings of the Third Eurographics Conference on Computational Aesthetics in Graphics, Visualization and Imaging*, 105–112.

Ritchie, G. 2007. Some empirical criteria for attributing creativity to a computer program. *Minds and Machines* 17(1):67–99.

Saunders, R., and Grace, K. 2008. Towards a computational model of creative cultures. In *AAAI Spring Symposium: Creative Intelligent Systems*, 67–74.

Saunders, R.; Gemeinboeck, P.; Lombard, A.; Bourke, D.; and Kocaballi, A. B. 2010. Curious whispers: An embodied artificial creative system. In *Proceedings of the 1st International Conference on Computataional Creativity*, 100–109.

Shannon, C. E. 2001. A mathematical theory of communication. *ACM SIGMOBILE Mobile Computing and Communications Review* 5(1):3–55.

Ventura, D. 2016. Mere generation: Essential barometer or dated concept. In *Proceedings of the 7th International Conference on Computational Creativity*, 17–24.

Whorf, B. L.; Carroll, J. B.; Levinson, S. C.; and Lee, P. 2012. *Language, Thought, and Reality: Selected Writings of Benjamin Lee Whorf*. MIT Press.

Wiggins, G. A. 2006. A preliminary framework for description, analysis and comparison of creative systems. *Knowledge-Based Systems* 19(7):449–458.

An HBPL-based Approach to the Creation of Six-word Stories

Brad Spendlove, Nathan Zabriskie, and Dan Ventura
Computer Science Department
Brigham Young University
Provo, UT 84602 USA
brad.spendlove@byu.edu, nathanzabriskie@byu.edu, ventura@cs.byu.edu

Abstract

Six-word stories are a subgenre of microfiction that presents significant challenges to authors. It is difficult to express a story with so few words, let alone an impactful or elegant one, but the best examples in the genre transcend mere storytelling to elicit an emotional response from readers. Six-word stories are an interesting and as-yet-unexplored space in the field of computational creativity. We present a system description of MICROS, a computationally creative system that generates six-word stories by making use of a hierarchical Bayesian approach. We detail how MICROS generates, evaluates, and refines stories; present stories generated by MICROS; and report a preliminary evaluation of its results.

Introduction

Six-word stories are a subgenre of microfiction (short stories composed of 1000 words or less). This restrictive word count might fool the casual observer into thinking six-word stories are simple to write, but this is not the case. With so few words to work with, the author does not have time to build up complicated worlds and characters and must instead seek to create the maximum possible impact upon the reader with every word.

"For sale: baby shoes, never worn.", a famous six-word story often attributed to Hemmingway, aptly demonstrates the potential impact of this genre of microfiction. In just a few words, the reader is given a glimpse into a larger story world before being left to themselves to imagine what else might be happening "around" the text.

Authors must display a great deal of creativity to paint these mental pictures within such a small frame. They use uncommon or unconventional grammatical structures both to fit their story into the six-word limit and to further amplify the effects of their words. As a result, the forms their stories take are often unique, both grammatically and in the semantic relationships between words.

The marriage of rigid length requirements and fluid grammatical structures makes proceduralizing the writing of six-word stories particularly difficult. Previous research has focused on computational generation of short stories and poetry but not specifically on six-word stories.

Narrative systems such as MEXICA (Pérez y Pérez and Sharples 2001), STellA (León and Gervás 2014), and Fabulist (Riedl and Young 2006) create stories by tracking characters' motivations, states of being, and locations as they perform actions to progress the plot. The story artifacts these systems output take the form of sequences of simple action phrases built from a fixed set of rules.

These systems do succeed in creating interesting stories, but the methods they use to achieve that success are often not applicable to six-word stories. For example, MEXICA will identify moments of low-interest and add new actions, such as a murder, to add tension to the narrative. Conversely, six-word stories are improved by fitting more story into the same number of words; they do not have the luxury of adding length to improve a story.

One approach to writing six-word stories would be to take a longer story, such as the output of an existing story generation system, and reduce it to six words. Such a reductive approach, however, seems unlikely to result in an interesting six-word story. Although a sequence of actions may make for an interesting story, taking a single action out of its context to fit within six words would likely result in an uninteresting or nonsensical story. This precludes a six-word story generator from working with existing story generators, such as serving as a module in a collaborative system like Slant (Montfort et al. 2013).

In fact, six-word stories rarely communicate a full narrative. Instead, they spend their limited lexical resources inviting the reader's imagination to fill in the story sketched out by the text. The best stories harness that imaginative leap to elicit emotions in the reader as well. In this way, six-word stories are more similar to poetry than narrative prose. Many previously developed poetry generators operate with a similar goal: to create poems that instill a certain feeling in the reader or deal with a specific topic.

The system described in (Colton, Goodwin, and Veale 2012) evokes a specific mood by modifying similes taken from an existing corpus to shift them towards the desired emotion. The modified phrases are then placed into user-defined templates to create a full poem. Other systems such as (Toivanen et al. 2012) swap out words in existing poems to change the topic of a poem. These systems capture the relationships between words and choose words with specific relationships to create a poem. This focus on semantics instead of grammar is a common thread between generating poetry and generating six-word stories.

Computationally creative systems that deal with humor and wit face similar challenges of brevity and semantic precision (Binsted and Ritchie 1994; Oliviero and Carlo 2003).

We propose approaching the creation of a story by sampling from a distribution over the space of all possible stories. Thought of this way, story creation can be understood using the framework of Hierarchical Bayesian Program Learning (HBPL) (Lake, Salakhutdinov, and Tenenbaum 2015). This has previously been demonstrated as a viable approach to computational creativity (Bodily, Bay, and Ventura 2017), and while one may be tempted to argue that it is not suitable for producing large, complex stories, it is a very general, useful framework in which to consider many possible operationalizations for different CC tasks. In particular, here we adopt this framework to describe an approach to creating six-word stories, and present an implementation of the framework that we call MICROS.

An HBPL View of Story Writing

Consider the set $\mathscr{W}$ of all possible words, and let a story $S = w_1, w_2, \ldots, w_n$ be a sequence of n words w_i with $w_i \in \mathscr{W}$. Then a probabilistic approach to the problem of story creation imposes a distribution $p(S)$ over the set $\mathscr{S}$ of all possible stories S. That is, the joint distribution $p(S) = p(w_1, w_2, \ldots, w_k)$ must be computed. If this is possible, creating a story means simply sampling from $p(S)$. Of course, for stories of any length, this distribution is likely to be intractable to compute, and thus typically some simplifying assumptions are made that allow the joint distribution to be factored in some way. HBPL suggests that there are domain-specific factorizations that both simplify the computational demands of such an approach and that exhibit explanatory power as well. For example, in the case of story writing, one might consider a factorization such as

$$p(S) = p(\kappa) \prod_{i=1}^{\kappa} p(m_i|\kappa) p(C_i|i, m_i) p(R_i|C_1, ..., C_{i-1})$$

where $p(\kappa)$ models the number of chapters in S; $p(m_i|\kappa)$ models the number of paragraphs for the ith chapter for a story with κ chapters; $p(C_i|i, m_i)$ models the ith chapter with m_i paragraphs; and $p(R_i|C_1, ..., C_{i-1})$ models the relation of the ith chapter to the previous chapters.

There are several things to note about this formalism. First, it is clearly hierarchical, and the subdistributions can be further factorized until (hopefully) they become tractable to compute. Second, the formalism says nothing about how the individual distributions of the factorization should be computed. Third, the form of the factorization imposes structure on both the story generated and on the process used for its generation; e.g., in the factorization shown here, Chapter 1 cannot depend on anything in chapters that follow it, and it must be written before any following chapters can be. Factorizing the joint distribution in a different way will admit other story and process structures. If the structure imposed by the factorization is "correct", and if the individual distributions can all be modeled tractably, the result should be "good" stories.

For the case of microfiction stories, much of the hierarchy collapses, of course, and the most complete factorization comes from an application of the chain rule[1]:

$$p(S) = p(w_1, w_2, \ldots, w_n) = \\ p(w_{i_1}) p(w_{i_2}|w_{i_1}) \ldots p(w_{i_n}|w_{i_1}, w_{i_2}, \ldots, w_{i_{n-1}})$$

where $i_j \in [1, n]$ and $i_j \neq i_k$ unless $j = k$, so that this represents a general version of the chain rule that admits any possible permutation of word order dependency.

Alternatively, we can consider a joint conditioned on some input $E = e_1, e_2, \ldots, e_m$, where $e_i \in \mathscr{W}$:

$$p(S|E) = p(w_1, w_2, \ldots, w_n|e_1, e_2, \ldots, e_m)$$

In what follows, we refer to a particular factorization of the joint as a story *format* and to the individual distributions (factors) as story *primitives*. For now, we have restricted the MICROS system to implementing a single format with six primitives, as follows:

$$p(S|E) = p(w_1, w_2, w_3, w_4, w_5, w_6|e_1) = \\ p(w_5|e_1) p(w_6|w_5, e_1) p(w_4|w_5, w_6, e_1) * \\ p(w_1|w_4, w_5, w_6, e_1) p(w_2|w_1, w_4, w_5, w_6, e_1) * \\ p(w_3|w_1, w_2, w_4, w_5, w_6, e_1)$$

which, given additional independence assumptions that we've made, can be simplified to this:

$$p(w_1, w_2, w_3, w_4, w_5, w_6|e_1) = \\ p(w_5|e_1) p(w_6|w_5) p(w_4|w_5) * \\ p(w_1|w_5, w_6) p(w_2|w_1, w_5, w_6) * \\ p(w_3|w_1, w_2, w_5, w_6)$$

The MICROS System

The operationalization of the format we selected for our system's stories consists of three background nouns (which we call *punchies*) that set the stage for the story and an article/subject/verb phrase that represents the action in the story. That is, w_1, w_2, w_3 are the punchies, w_4 is an article, w_5 is the subject, and w_6 is the verb. A simple example story in this format is, "Gun. Mask. Note. The teller screams."

Our system takes as input E a subject noun e_1 and returns a story $S = w_1, w_2, w_3, w_4, w_5, w_6$ in the format described above. Rewriting that format now using the word types just discussed gives

$$p(punchy_1, punchy_2, punchy_3, \\ article, subject, verb|noun) = \\ p(subject|noun) p(verb|subject) * \\ p(article|subject) p(punchy_1|subject, verb) * \\ p(punchy_2|punchy_1, subject, verb) * \\ p(punchy_3|punchy_1, punchy_2, subject, verb)$$

[1]other less complete factorizations can, of course, also be considered, e.g., $p(S) = p(w_{i_1}, \ldots, w_{i_{\lfloor n/2 \rfloor}}) p(w_{i_{\lceil n/2 \rceil}}, \ldots, w_{i_n}|w_{i_1}, \ldots, w_{i_{\lfloor n/2 \rfloor}})$.

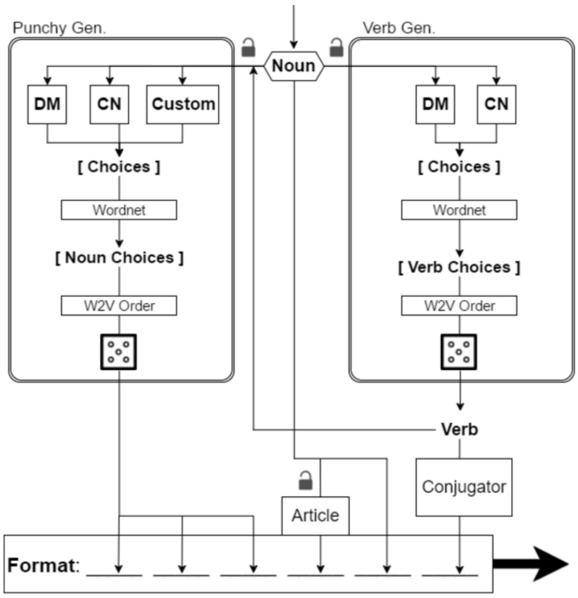


Figure 1: MICROS' six-word story generator.

These six primitive distributions are built by a generator module that then samples them to return an initial story. That story is scored with an evaluator module and passed to a refiner module that searches for higher scoring stories by resampling (some of) the primitive distributions. The refiner's output is the final six-word story artifact.

Generator

At a high level, the generator uses the input noun as the subject and then chooses a verb using the given subject. It then chooses three punchies using the subject and verb. Finally, the article is chosen, the verb is conjugated to the present tense, and the completed story is returned from the generator. Figure 1 is a diagram of this process, and the following sections detail how each primitive is constructed.

Subject and Article Modeling The subject distribution $p(subject|noun)$ is trivial to construct as all of the probability mass is placed on the input noun e_1. Sampling this distribution will therefore always return e_1 as the subject.

The article distribution $p(article|subject)$ is also simple to construct. MICROS assigns a probability of 0.5 to "the" and 0.5 to the appropriate form of "a/an" based on the chosen noun.

Verb and Punchy Modeling In order to capture the appropriate semantic relationships, the verb distribution $p(verb|subject)$ and the three punchy distributions (each conditioned on the subject, the verb, and any preceding punchies) are computed using various resources that relate words to one another. The method by which these primitive distributions are computed is similar, so we will describe them in parallel.

Both primitive generators start by collecting a set of words related to the words on which the primitive is conditioned. This is represented in Figure 1 with the inputs to the generators (the Noun for the verb generator, and the Noun and Verb for the punchy generator) being first fed into different modules that find related words. Two of these modules, labeled "DM" and "CN", query APIs for words that are related to the input words in different ways. The punchy generator has an additional related-words module called "Custom" that will be described below.

The module labeled "CN" queries ConceptNet (Speer and Havasi 2012), which stores English terms and their relationships to one another. ConceptNet defines 28 different relations that represent the ways that terms can be related. MICROS' verb generator draws on the "CapableOf", "UsedFor", and "Desires" relations while the punchy generator gathers candidate words from the "HasSubevent", "Causes", "HasPrerequisite", and "UsedFor" relations.

ConceptNet is populated from a variety of sources including semantic knowledge bases, dictionaries, and crowdsourced data. Probably due to these disparate sources, the terms and relations may be populated sparsely, are not comprehensive, may include duplicates, and sometimes reflect obscure senses of words. For example, although the word "actor" has a highly populated "CapableOf" relation, the related terms include both useful phrases like "act in a play" and "star in a movie" and oddities such as "cake on makeup" and "milk a part".

Our generator also draws from a knowledge base called Datamuse [2], represented by the "DM" module in Figure 1. Datamuse takes a word as input and returns a list of words that match certain constraints. MICROS uses the constraint called "triggers", which relates a word to other words that are statistically likely to be seen in the same literary context. For example, words that are triggered by "baby" include "boomer" and "doll".

This example also reveals a limitation with DataMuse. "Boomer" and "doll" are both words that are statistically likely to appear with "baby" in text, but they are semantically related to "baby" in radically different ways.

The final source of word relationships MICROS uses to build primitive distributions is word2vec (Mikolov et al. 2013). Word2vec is a natural language processing model that embeds words in a vector space. Using a large corpus, word2vec constructs a high-dimensional space in which each word in the corpus is represented as a vector from the origin to an associated point.

It has been demonstrated that the geometry of this vector space can represent semantic relationships between words. For example, the vector "king" minus the vector "man" plus the vector "woman" results in the vector "queen". The ability to perform vector operations on words is powerful, and the full potential of these types of approaches is still being explored.

MICROS uses word2vec to create a custom relation which we call "caused by". This relation (represented in Figure 1 by the "Custom" module) is formed by calculating the vector between example word pairs such as "eat, hunger", "drink, thirst", and "scream, fear." The vectors be-

[2] http://www.datamuse.com/api/

tween the two words in each pair are averaged together to create a relation vector. That relation vector is added to the vector of an input word, yielding a point in the word2vec space. Word vectors that are close to that point are likely to be "caused by" the input word. For example, for the input verb "help" our "caused by" custom relation returns words such as "humanitarian", "concern", and "compassion".

The outputs of these modules (DM and CN for the verb generator and DM, CN, and Custom for the punchy generator) are collected as a list of related words, which we will refer to as "choices". The choices are then filtered to include only single words of the correct part of speech, labeled in Figure 1 as "Verb Choices" and "Noun Choices", respectively. This is accomplished by querying WordNet, a large lexical database (Miller 1995).

WordNet stores words in semantically related groups called synsets that contain properties such as part of speech. MICROS also uses WordNet to discard duplicate choices that are not identical strings by taking advantage of a function called "morphy" that reduces words to their base forms, e.g. singular nouns or infinitive verbs.

Once the choices are stripped down to unique, single words of the correct part of speech, MICROS employs word2vec to score them by their similarity to words previously sampled from any primitives further up the hierarchy. Thus, the verb generator scores its choices compared to the noun, and the punchy generator scores its choices compared to the noun and the verb. Each choice word is embedded as a word2vec vector, and the cosine similarities between that vector and the vector representation of each of the preceding primitive words are summed to give a score. The lists of choices are sorted according to these scores, in ascending order, and are labeled "W2V Order" in Figure 1.

Choices that are very dissimilar from the input words are likely to be unrelated or incoherent, so the bottom fifth of the ordered list is discarded. As an example, when punchy choices are generated for the subject/verb pair "cowboy rides" the most similar words include "outlaw", "bandit", and "desperado", and the least similar words include "parks" and "symposium".

After the most dissimilar words are discarded, the generator builds the distribution $p(w_i|\dots)$ by assigning a probability to each word in the choice list, proportional to its word2vec similarity score. The generator then samples from this distribution and inserts the chosen word into the story (represented by a dice icon in Figure 1).

As seen in Figure 1, the output of the verb generator is both used as input to the punchy generator and fed into a conjugator module. Conjugation, article agreement, and later pluralization are all accomplished using the pattern.en Python library[3]. This library uses pattern matching rules and exceptions to do simple grammar tasks and is much faster than a more comprehensive dictionary-style lookup.

The punchy generator takes the verb and noun as input and computes the three punchy primitives, which differ only in that choice words that have been sampled from previous draws from a punchy primitive are assigned a probability of 0 so that punchies cannot be repeated. The punchies sampled from the three punchy primitives are slotted into the story as the first three words, followed by the article, input noun, and conjugated verb. This forms the six-word story output by the generator.

[3]https://www.clips.uantwerpen.be/pages/pattern-en

Resampling After MICROS generates $p(S|E)$ it can selectively resample primitive distributions in order to create a new story. This is done by "locking" certain words and resampling the remaining words, which mutates the original story. For example, to change the story "Humanity. Adventure. Rider. The wizard quests.", all of the words except "adventure" could be locked so only the punchy primitive $p(punchy_2|$'humanity', 'wizard', 'quest'$)$ would be resampled. The lockable primitives are represented in Figure 1 by a padlock icon.

Before sampling each primitive distribution, the generator first checks to see if that word is locked. If it is, then the locked word remains w_i rather than being (re-)sampled from the corresponding primitive distribution.

The factorization hierarchy described above affects how primitives are generated with locks. If a primitive higher in the hierarchy is unlocked and resampled, all locks for primitives lower than it will be ignored and the system will build new distributions for those primitives and sample from them. Otherwise, the system would keep punchies that are unrelated to the current verb, for example, violating the semantic relationship between the primitives. The generator's ability to mutate stories one primitive at a time will become relevant in the discussion of refinement below.

Caching Populating each primitive distribution with related words from ConceptNet and Datamuse requires network API calls that are slow to execute. In order to minimize API requests, the generator caches distributions for later reuse when the story is mutated by the refiner module.

The verb primitive distribution $p(verb|subject)$ is built once and cached. Because the subject never changes during MICROS' execution, this cached distribution can be resampled for free as many times as necessary. Punchy primitive distributions are cached on a per-verb basis and may or may not be resampled during the execution of the refiner.

Although building each distribution takes only a few seconds, this process may occur many times as verb primitives are unlocked and new punchy primitives are built. Our testing showed that MICROS' caching scheme saves an average of 100 seconds over the system's 10 minute runtime.

Exploration

The preceding section described how the generator builds a distribution $p(S|E)$ from which to sample a story S. Once the first story for a given input has been sampled, MICROS evaluates it and refines it to create progressively better stories by modifying the primitive distributions. This section details the operation of the evaluator and refiner modules used to complete this process.

Evaluation MICROS' evaluator is based on skip-thought vectors (Kiros et al. 2015), which are mappings of natural language sentences to a high-dimensional vector space, with

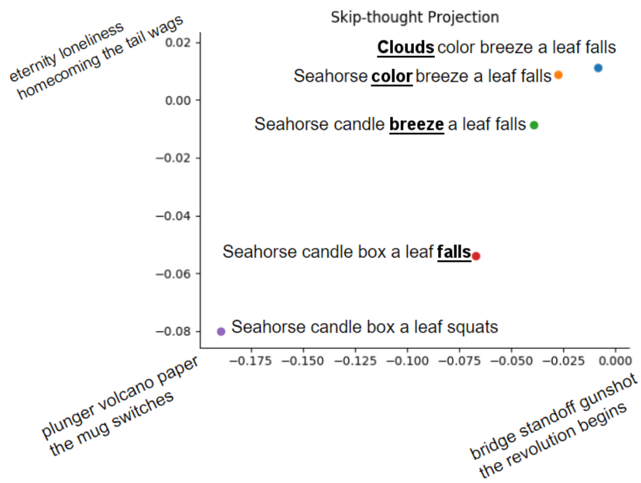


Figure 2: Example of skip-thought scores for progressively more coherent six-word stories. The underline indicates which word changed as the story mutates one word at a time from "Seahorse candle box a leaf squats" to "Clouds color breeze a leaf falls".

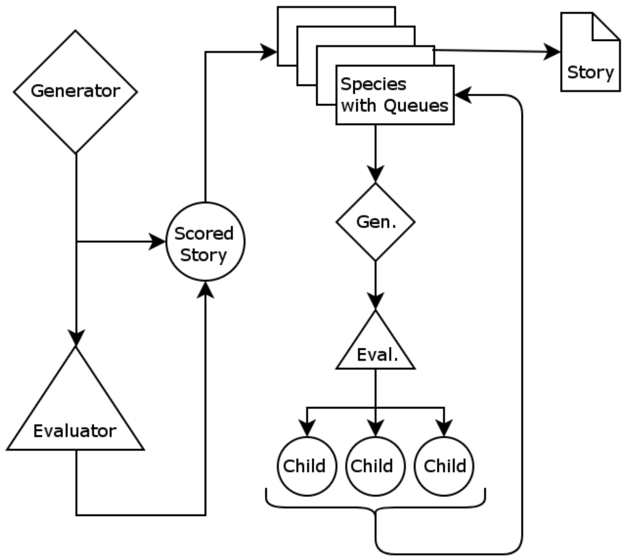


Figure 3: MICROS' refinement process.

semantically similar sentences mapping to similar vectors. In this way, skip-thoughts can be thought of as a sentence-level version of word2vec: while vectors in word2vec represent words, skip-thought vectors represent sentences.

Each time the evaluator receives a story, it first encodes it as a vector using a skip-thought encoder that was pre-trained on a large corpus of novels. It then projects this vector onto a two-dimensional plane defined by two axis vectors.

These axis vectors are calculated by subtracting the vector representation of a "bad" six-word story from the vectors of two "good" stories, all of which are hardcoded and in the same format as the generated stories. The bad vector, which forms the origin of the two-dimensional plane, is a nonsensical collection of six words while the good vectors are human-written, cohesive stories. In our system, the bad story is "Plunger. Volcano. Paper. The mug switches." and the two good stories are "Bridge. Standoff. Gunshot. The revolution begins." and "Eternity. Loneliness. Homecoming. The tail wags."

By defining the axes in this way, a story that projects to a positive coordinate will be more coherent than a story that projects to a negative coordinate. Our initial experiments showed that these axes give scores that reflect coherence reasonably well. Figure 2 shows the results of an experiment in which we took a randomly generated story and changed one word at a time to make it increasingly coherent. As can be seen, after each word change (increasing coherence), the resulting story maps to a vector with increasingly positive coordinates on the plane.

MICROS' evaluator scores each story using Manhattan distance from the origin point on the plane, yielding more positive scores for more coherent stories.

Refinement Using the generator and evaluator described above, our system refines stories through a branch-and-bound style search. Figure 3 shows how this process works.

At its most basic level, the refiner takes generated stories and places them in a priority queue according to their score. The queue is initially populated with the first generated story for the given input. At each iteration, the system dequeues the story with highest priority, saves it if it has the highest score seen so far, and mutates it several times by locking all but one of the story's words and resampling $p(S|E)$.

The resampled "children" stories are then scored and placed back into the queue to prepare for the next search iteration. By repeating this process for many iterations, the system searches for stories with higher scores. This cycle is represented in Figure 3 by the loop from the queue, through the generator and evaluator, and back into the queue.

This refinement method can unfairly favor some stories over others. Because a format's primitives exist in a hierarchy, changing the verb primitive of a story causes all of the punchy primitives to be regenerated because they are lower in the hierarchy. If MICROS mutates the verb of a story, the resulting child will differ from its parent by its three punchies and its verb. Because the child has changed so dramatically, it is likely to be less coherent and thus score lower than its parent, which may have already gone through many refinement loops with the same verb. If stories with newly-generated verbs are forced to compete against stories with already-refined verbs, they will likely continually be given low priority and never reach the front of the queue.

To mitigate this unfairness, the system speciates stories based on their verb, maintaining several species during the refinement process. A species maintains its own priority queue and tracks the highest scoring story it has seen so far. At each iteration, every species dequeues the story at the front of its queue and mutates it to create a new group of children. MICROS then scores each child story and inserts it into the species queue that matches its verb, creating a new species if the verb has not yet been seen.

Because transitive verbs do not fit grammatically into our format, if a story is generated with a transitive verb its species is marked as "transitive" and is never dequeued or iterated upon. To determine a verb's transitivity, we query the Oxford Dictionary API[4]. This API allows only a limited number of requests per month, but because our species each represent a unique verb, we only need to query the API once per species. Although they will never be iterated upon, preserving transitive verb species allows MICROS to avoid redundant API calls.

At the end of each iteration, if a species has not improved its maximum score for a preset number of iterations, it is considered stagnant and will no longer be dequeued to produce children. Once all species are stagnant, MICROS performs a weighted choice of the top scoring stories of every intransitive species and returns it as the output of the refinement process.

Our system performs this final randomization rather than automatically returning the highest scoring story overall because the skip-thought evaluator can favor some noun/verb combinations, consistently assigning them high scores run after run. Despite this favoritism, the stories in the top-scoring species are often of comparable quality. By performing a weighted random choice, our system gives less-favored combinations a chance to be chosen as output.

The selected story is prepared for final output by pluralizing the punchies. To accomplish this, MICROS uses the Oxford Dictionary API to check whether the punchies are mass or proper nouns. If they are not, they are pluralized with pattern.en. Once that is complete, the story is output as the final artifact.

Results & Analysis

Six-word stories, like all art, can only be judged subjectively. Each reader will come to their own conclusion about the quality of a given story, and that estimation may change over time. Thus, our discussion of our system's results will naturally be biased based on our own tastes. Such bias cannot be removed, so instead we will briefly explain what we think makes a good six-word story.

Our first criteria is *coherence*—does the story make logical sense? Next, we consider *impact*—does the story elicit an emotional response? Finally, we consider a story's *subtlety*. The best six-word stories tell their stories without explicitly stating the story's topic, mood, or even its central action. The words in a subtle six-word story all semantically "point" to a story but either do not tell it explicitly or tell only a portion of it. This allows the reader's mind to fill in the gaps, resulting in a deeper and more interesting story.

With these criteria in mind, we turn to an analysis of our system's results and performance. All examples given in the following subsections were created by MICROS. See the Appendix for more results.

[4]https://developer.oxforddictionaries.com/

Successes

Our system generally succeeds at coherence; the majority of its artifacts fulfill this criteria and evoke at least a vague narrative. The examples "Schools. Institutes. Honors. A bachelor graduates." and "Redemption. Crimes. Chaos. A policeman escapes." typify the coherence of our system's artifacts; most of the words are at least loosely associated and paint a hazy picture in the mind of the reader. A story's skip-thought score tends to increase as its coherence improves. As a result, the evaluator favors and selects coherent stories.

Our system's artifacts only rarely have an impact on the reader. In the majority of cases, their coherence does not serve to tell an interesting story. Occasionally, an artifact's elements will combine to portray a fairly interesting narrative, but even those stories are still vague or slightly nonsensical. "Injustice. Ambitions. Minnesota. A farmer emigrates." and "Cowardice. Injustice. Motive. A hero stands." tell slightly interesting stories, and "Depravity. Misfortune. Hysteria. A clown laughs." succeeds in eliciting fear or dark humor in the reader. MICROS' stories are impactful when they are highly coherent and, by chance, the generated primitives are emotionally charged words.

Shortcomings

Although the majority of MICROS' artifacts are coherent, some are not due to the independence assumptions made by the format (factorization). Although this can yield nouns and verbs that don't make sense together, it occasionally results in comical or punny combinations such as "A wizard potters." or "A mechanic brakes". However, these punny stories often lack coherence because the generator can't choose punchies to match untraditional subject/verb pairs, such as "Immortality. Berserkers. Adventures. A wizard potters."

Our reliance on WordNet to identify verbs also causes problems because it contains every conceivable sense of a word, even archaic or obscure ones. For example, one of its synsets for the word "harlequin" lists it as a verb (which is defined as "[to] variegate with spots or marks"). Thus, if "harlequin" was returned from a relation, WordNet would identify it as a verb, even though using it as one in a story would likely confuse the reader.

WordNet is fast but clearly not the best way to determine the parts of speech in a phrase. However, even if a sophisticated parser—such as the Stanford Parser (De Marneffe, MacCartney, and Manning 2006)—was used to identify the relationship's parts of speech more accurately, the words' meanings would still be unknown. The example "Seclusion. Pregnancy. Love. A baby sleeps." demonstrates this. Even though the three punchies are all individually related to either "baby" or "sleep", the story makes no sense because the system has no way of knowing how they are related or how they should be used.

ConceptNet fails to provide such deep semantic relationships because its relations for each word are not separated by sense. A richer semantic database could represent relationships between specific word senses, instead of conflating all those senses into single terms. This would allow deeper understanding of the relationships between words while retaining ConceptNet's easy-to-search structure of relations.

The most difficult criteria for a six-word story to fulfill is subtlety. Even human writers struggle to write good stories that are just subtle enough to be interesting without being vague or illogical. Our system cannot compete at this level. Its most impactful stories still consist of words that are all directly related to the noun or verb.

MICROS' method of constructing primitives precludes it from achieving subtlety; each primitive is generated by selecting words that are directly related to parent words. If instead primitives were generated such that they all related to a separate concept that was not itself a primitive, the resulting story could perhaps approach that latent concept subtly.

Finally, MICROS makes occasional pluralization or conjugation mistakes such as in the story "Destructivenesses. Aristotles. Questions. A philosopher thinks." These mistakes are not unexpected and are due to the limitations of the pattern.en Python library. Although fast, pattern.en is limited to pattern matching and does not explicitly use the rules of English grammar. However, it is a good example of how such rules can be simplified to work in most cases. "Good enough" solutions like pattern.en are useful to creative computer systems where generation speed is often more important than correctness.

Community Evaluation

Evaluation by a community is the ultimate metric of artifacts' value, and we are pleased to report that MICROS performed better than some human writers in a real world environment. In order to test the results of our system in the wild, we submitted one MICROS-generated story a day to the /r/sixwordstories community on Reddit for a week. Each submitted story was freshly generated by MICROS on the day of submission and posted without curation.

Reddit is a social media platform that allows users to "upvote" content they like and "downvote" content they don't. The /r/sixwordstories subreddit has over 29,000 subscribers and one-to-two dozen submissions daily. Importantly, the stories posted on the subreddit are written by average users, who may be amateur writers at best. The typical best post on any given day will have 40–100 points and posts with less than 6 points are common. (A post's points are basically its upvotes minus its downvotes, to a minimum of 0.)

Of the seven stories we posted, two have 0 points, three received no votes, and two have positive scores: 5 and 7, respectively. Although those numbers may seem low, they do outscore other posts (presumably) written by humans. The story "Companionship. Youths. Fulfillment. The teacher cares." scored 5 points, which was higher than 5 of the 10 other stories posted that day. "Poverty. Retribution. Heroes. A villain acts." scored 7 points, outscoring 5 of the 12 other posts that day. These results are encouraging and show that MICROS can compete in the six-word story community, at least among amateur writers.

Conclusion & Future Work

MICROS represents an initial foray into the creation of microfiction using a novel HBPL-based approach. Although its results are often bland and never rise to the level of truly great writing, MICROS' approach to creative computation could serve as both an example and jumping off point for future research in computational creativity.

Our HBPL-based approach to formally defining the format of a creative artifact provides a convenient way to describe stories and poetry. It is more descriptive, for example, to define MICROS' six-word story format as being composed of three punchy primitives followed by an article primitive, subject primitive, and verb primitive than to simply list the parts of speech.

This approach of intentionally choosing factorizations of $p(S)$ to give a desired structure to and relationship between the different parts of a creative artifact could easily be applied in other story or poetry generation contexts to provide the system with information about how each word or phrase of the piece should relate to the others.

Conversely, MICROS could incorporate a module that programatically extracts factorizations of $p(S)$ from human-written text. A system that can identify primitives and learn the semantic relationships between them would allow a creative system to generate more varied and novel artifacts while retaining the semantic richness of the underlying structure. For example, MICROS could incorporate elements of the system described by (Toivanen et al. 2012) to operate on a corpus of existing stories, analyze the semantic relationships between the words that comprise them, and sample new stories from the learned format instead of replacing words into stories directly. This would allow the replacement words to not only relate to a topic but also to the other words in the story as dictated by the format.

More sophisticated and nuanced primitives, whether designed by researchers or extracted from existing text, could allow MICROS to generate six-word stories with high emotional impact. The words in the famous "baby shoes" story all relate to one another in a deep semantic way, and encapsulating those semantics into a factorization of $p(S)$ would enable the system to generate equally compelling stories.

Although MICROS currently only operates in the domain of writing, our approach is powerful because it is domain agnostic. The input E in $p(S|E)$ does not necessarily need to match the domain of S. Similar to how a human mind can be inspired to write a poem by a beautiful view, future work could allow MICROS to create six-word stories by sampling from a distribution conditioned on music or images. Designing factors of $p(S|E)$ that connect disparate domains could be an interesting avenue for future research.

By building systems that harness the rich semantic connections between inspirational and artistic domains as well as the equally-rich connections between the individual elements that comprise a creative artifact, we can further approximate human creativity. The MICROS system we have presented in this paper represents the first steps toward that goal in the domain of six-word stories, and the descriptive formalism we have adopted provides guidance for the designers of future creative systems by framing computational creativity in a standard formal structure.

Appendix

This appendix contains various six-word story artifacts created with MICROS. These stories were generated sequentially and are presented without curation.

Mirages. Injustice. Adventures. A ninja revenges.
Immortality. Berserkers. Adventures. A wizard potters.
Devotion. Honors. Alphas. An undergraduate majors.
Vocations. Thoughts. Schools. A philosopher teaches.
Vengeance. Hordes. Injustice. A soldier battles.
Necessities. Wellbeings. Achievements. A student begins.
Jealousy. Children. Immortality. A baby rattles.
Parties. Retribution. Parliaments. A manager resigns.
Injustice. Foes. Despair. A farmer rises.
Humanity. Devotion. Retribution. A hero stands.
Guardians. Followups. Indifference. A reporter replies.
Injustice. Acting. Regimes. A woman falls.
Devotion. Charlottes. Seconds. A queen plays.
Prisons. Courts. Justices. The lawyer comments.
Bloodlust. Diseases. Fulfillments. A surgeon parts.
Forgiveness. Weekends. Retaliation. A reporter replies.
Terrors. Actresses. Love. An actor shows.
Journeys. Terrors. Torment. A horse travels.
Personalities. Meltdowns. Desires. A plumber leaks.
Cavalries. Hardship. Regiments. A soldier drives.
Loneliness. Troupes. Champions. A dancer wells.
Families. Companionship. Fasts. A dog eats.
Terrors. Aces. Adventures. A monster cards.
Families. Dyings. Reigns. The king peoples.
Consciousnesses. Reigns. Sweden. A king groups.
Tournaments. Substitutes. Feuds. A wrestler matches.
Parties. Votes. Mates. A senator resigns.
Failure. Motives. Generations. A computer powers.
Talents. Partners. Concerns. A celebrity letters.
Destinies. Afterlives. Hysteria. A monster appears.
Loneliness. Affection. Misfortune. A student lunches.
Loyalty. Demoralizations. Souls. A band disbands.
Epics. Folks. Impunity. A poet rhymes.
Affection. Grandsons. Perfection. A politician plays.
Soccer. Appearances. Feuds. A wrestler competes.
Killings. Duties. Crimes. A policeman soldiers.
Elation. Wingers. Detriment. A writer rights.
Companionship. Injustice. Chases. A woman hunts.
Sights. Evil. Samurais. The ninja eyes.
Humanity. Redemption. Depredations. A hunter preys.
Rebirth. Loneliness. Adventures. A hobo bums.
Abhorrence. Wraiths. Revenge. A pirate bilges.
Academies. Graduation. Guys. A professor smarts.
Humanity. Causes. Necessities. A hero stands.
Contentment. Empresses. Sorrow. A queen consorts.
Weights. Collisions. Failure. A mechanic brakes.
Riders. Mankind. Desires. A horse races.
Hinds. Desires. Destinies. A hunter tails.
Adventures. Wells. Mafias. A detective gangs.
Humanity. Devotion. Templars. A knight duels.
Companionship. Bravery. Strikes. A coach plays.
Poverty. Everlastings. Retribution. A woman acts.
Fates. Tragedies. Adventures. A ninja revenges.
Soccers. Leagues. Childishnesses. A wrestler teams.

References

Binsted, K., and Ritchie, G. 1994. A symbolic description of punning riddles and its computer implementation. *arXiv preprint cmp-lg/9406021*.

Bodily, P.; Bay, B.; and Ventura, D. 2017. Computational creativity via human-level concept learning. In *Proceedings of the 8th International Conference on Computational Creativity*, 57–64.

Colton, S.; Goodwin, J.; and Veale, T. 2012. Full-FACE Poetry Generation. In *Proceedings of the 3rd International Conference on Computational Creativity*, 95–102.

De Marneffe, M.-C.; MacCartney, B.; and Manning, C. D. 2006. Generating typed dependency parses from phrase structure parses. In *Proceedings of the International Conference on Language Resources and Evaluation*, volume 6, 449–454.

Kiros, R.; Zhu, Y.; Salakhutdinov, R. R.; Zemel, R.; Urtasun, R.; Torralba, A.; and Fidler, S. 2015. Skip-thought vectors. In *Advances in Neural Information Processing Systems*, 3294–3302.

Lake, B. M.; Salakhutdinov, R.; and Tenenbaum, J. B. 2015. Human-level concept learning through probabilistic program induction. *Science* 350(6266):1332–1338.

León, C., and Gervás, P. 2014. Creativity in story generation from the ground up: Non-deterministic simulation driven by narrative. In *Proceedings of the 5th International Conference on Computational Creativity*, 201–210.

Mikolov, T.; Chen, K.; Corrado, G.; and Dean, J. 2013. Efficient estimation of word representations in vector space. *arXiv* abs/1301.3781.

Miller, G. A. 1995. WordNet: A lexical database for English. *Communications of the Association for Computing Machinery* 38(11):39–41.

Montfort, N.; Pérez y Pérez, R.; Harrell, D. F.; and Campana, A. 2013. Slant: A blackboard system to generate plot, figuration, and narrative discourse aspects of stories. In *Proceedings of the 4th International Conference on Computational Creativity*, 168–175.

Oliviero, S., and Carlo, S. 2003. Hahacronym: Humorous agents for humorous acronyms.

Pérez y Pérez, R., and Sharples, M. 2001. MEXICA: A computer model of a cognitive account of creative writing. *Journal of Experimental & Theoretical Artificial Intelligence* 13(2):119–139.

Riedl, M. O., and Young, R. M. 2006. Story planning as exploratory creativity: Techniques for expanding the narrative search space. *New Generation Computing* 24(3):303–323.

Speer, R., and Havasi, C. 2012. Representing general relational knowledge in ConceptNet 5. In *Proceedings of the International Conference on Language Resources and Evaluation*, 3679–3686.

Toivanen, J.; Toivonen, H.; Valitutti, A.; Gross, O.; et al. 2012. Corpus-based generation of content and form in poetry. In *Proceedings of the 3rd International Conference on Computational Creativity*, 175–179.

Data-driven Design: A Case for Maximalist Game Design
Paper type: Position paper

Gabriella A. B. Barros[1], Michael Cerny Green[1], Antonios Liapis[2] and Julian Togelius[1]
1: Tandon School of Engineering, New York University, New York, USA
2: Institute of Digital Games, University of Malta, Msida, Malta
gabbbarros@gmail.com, mcgreentn@gmail.com, antonios.liapis@um.edu.mt, julian@togelius.com

Abstract

Maximalism in art refers to drawing on and combining multiple different sources for art creation, embracing the resulting collisions and heterogeneity. This paper discusses the use of maximalism in game design and particularly in data games, which are games that are generated partly based on open data. Using Data Adventures, a series of generators that create adventure games from data sources such as Wikipedia and OpenStreetMap, as a lens we explore several tradeoffs and issues in maximalist game design. This includes the tension between transformation and fidelity, between decorative and functional content, and legal and ethical issues resulting from this type of generativity. This paper sketches out the design space of maximalist data-driven games, a design space that is mostly unexplored.

Introduction

The unprecedented availability of digital data impacts most human endeavors, including game design. In particular, freely available data can be combined with procedural content generation (PCG) and computation creativity to create systems that can generate games (or game content) based on open data. We have previously identified such games as "data games" (Gustafsson Friberger et al. 2013).

This paper explores some of the aesthetic challenges, particularities and concerns associated with games that are created from data. We start from the idea that the use of data games is in many ways similar to notions in art such as collage, sampling, and remixing. We draw on content from many different sources, causing creative collisions between them. This lets us apply some of the same conceptual apparatus to study data games as has been applied to these types of art. We also start from a series of game generators we have created, collectively referred to as "Data Adventures". These generators create adventure games, such as murder mysteries, from open data from e.g. Wikipedia and OpenStreetMap. Our ongoing struggle with getting these generators to produce playable and interesting content from something as varied and occasionally unreliable as Wikipedia has illuminated both possibilities and pitfalls of this approach.

This paper is an attempt to explore the design space of maximalist data-driven games (and other data games) in order to form an initial understanding of it. It is also an attempt to systematize reflections from our own and others' attempts at creating such games. We address the following questions:

- What does it mean for games designed from/for data to be maximalist?
- What is the tradeoff between transforming data and staying true to the source in terms of generating games?
- What are the characteristics of game content that can be generated from data?
- For what purposes can data-driven maximalist games be designed and how does that affect their character?
- What new legal and ethical issues, including copyright issues and the potential for generating offensive, misinforming and biased content, are raised by this type of game design?

Data-driven design and data games

This age of data sharing (whether sharing is free or not) has certainly been advantageous to research in computational creativity. While computational creativity does not necessarily need to emulate human creativity (Pease and Colton 2011), freely available human-annotated data can be exploited as an inspiring set (Ritchie 2007) to any creative software. In natural language generation, Google N-grams have been exploited to identify analogies and similes (Veale 2014), corpora of phonetic information for all words have been exploited to generate jokes (Ritchie and Masthoff 2011), and books of a specific author have been used to generate stories typical of the genre (Khalifa, Barros, and Togelius 2017). In visual generation, crowd-sourced annotations of data were used to create image filters (Heath and Ventura 2016), while object recognition models based on deep learning of Google images was used to choose how generated 3D shapes would represent an object (Lehman, Risi, and Clune 2016). Similarly, deep learning from massive musical corpora was used to create new music (Hawthorne et al. 2017).

In the creative domain of games, on the other hand, similar approaches have been used to create different game components. Google's autocomplete function (which uses a form of N-grams) was used to discover names for enemies and abilities of a game character whose name was provided by the player (Cook and Colton 2014). In the same game, Google image search used discovered names to select images for these enemies' sprites. In other work, Guzdial and Riedl (2016) used Youtube playthroughs to find associations in the placement of level elements (e.g. platforms, enemies) which were used to generate levels for *Super Mario Bros* (Nintendo 1985). Patterns in *Starcraft II* maps (Blizzard Entertainment 2010) were learned through deep learning (Lee et al. 2016); these encodings were used to change the frequency of minerals in the map without the usual exploratory process of e.g. an evolutionary algorithm. To better coordinate the learning process of level patterns, a corpus of diverse games has been collected (Summerville et al. 2016).

While using existing game data —often annotated with human notions of quality— has been explored in computational game creativity (Liapis, Yannakakis, and Togelius 2014), most efforts perform minor adjustments to existing games. Game generators such as Angelina (Cook, Colton, and Pease 2012), A Rogue Dream (Cook and Colton 2014), and Game-O-Matic (Treanor et al. 2012) use data outside the game domain, enhancing their outcomes with human-provided associations (and content such as images). Even so, the core gameplay loop is simple: in Angelina, for example, the player performs the basic actions of a platformer game (e.g. jump, run); in A Rogue Dream the player moves along 4 directions and perhaps uses one more action. Gameplay in all these games is mechanics-heavy, relying on fast reactions to immediate threats rather than on high-level planning or cognitive ability. Many *data games* take an existing game mechanic and generate new content for that game from open data (Gustafsson Friberger et al. 2013; Gustafsson Friberger and Togelius 2013; Cardona et al. 2014). In some cases, such as the game *Bar Chart Ball*, a new game mechanic is added to an existing data visualization (Togelius and Gustafsson Friberger 2013). To play even simple data games, the player must have some understanding of the underlying data. Playing data games requires some mental effort, deduction or memory; not only dexterity.

While most data-driven game generation software focus on a simple and tight gameplay loop, there is considerable potential in using and re-using information outside of games to create more complex game systems and more involved experiences. We argue that data-driven game generation can allow for a new gameplay experience. Using the Data Adventures series of game generators as a concrete example, we articulate the tenets of maximalism in game design inspired by the art movement of the same name. Moreover, we discuss two possible dimensions of maximalist game design, and how it can start from the raw data on one end or from the gameplay experience on the other. Finally, we envision the potential uses and issues of maximalist game design.

Maximalism in data-driven design

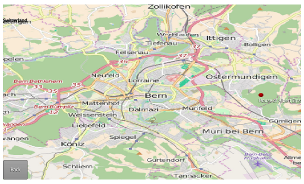
We are inspired by the notion of maximalism in the arts, rather than in the game design sphere. In music, for example, maximalism "embraces heterogeneity and allows for complex systems of juxtapositions and collisions, in which all outside influences are viewed as potential raw material" (Jaffe 1995). We similarly embrace the use of heterogeneous data sources as notes (i.e. the individual components) and melody (i.e. the overarching game or narrative structure) to produce a game as an orchestration of dissimilar instruments (Liapis 2015). In that sense, maximalism in data-driven design is likened with mixed media in art, where more than one medium is used. De facto, the heterogeneity of the data, its sources, and the people who contribute to its creation and curation will insert juxtapositions and collisions. This may not always be desired, and several catastrophic, inconsequential or seemingly random associations should be redacted. However, the "grain" of data-driven design (Khaled, Nelson, and Barr 2013) is built on the collision and absurdity of different elements that find their way into the game.

It should be noted that maximalism in the artistic sphere refers to materials or identities of elements within an image, song, or novel. We refer to maximalist game design in that sense, focusing on how game elements originating from different data sources (or transformed in different ways) are visualized, combined and made to interact together, thus not directly opposed to minimalist game design. Nealen, Saltsman, and Boxerman's (2011) minimalist game design encourages removing the unnecessary parts of the design, highlighting the important bits. Sicart's approach (2015) refers to the game loop; minimalist games have a simple core game loop which is largely unchanged throughout the game. Sicart uses *Minecraft* (Mojang, 2011) as an example where the simple core loop gather→craft→build that remains relevant and unchanged (except from the specific materials worked) throughout the game.

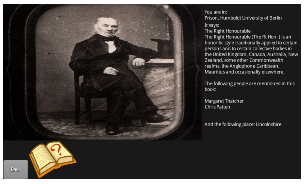
A data-driven game, maximalist in the artistic sense, can also be minimal in the gameplay loop sense. Data Adventures (Barros, Liapis, and Togelius 2015) has a simple core gameplay loop of traveling to a new location, talking to a non-player character in that location, learning the clue for the next location. Games that we would define as maximalist on the design sense, on the other hand, have the broadest mechanics of options for solving a problem — e.g. killing a dragon with stealth, magic, followers, swords, fists, poison, etc. in *Skyrim* (Bethesda, 2011) — or subsystems that are so elaborate or numerous that the player becomes unable to distinguish a core game loop — e.g. the diverse driving, shooting, spraying, running, etc. minigames in *Saints Row IV* (Deep Silver, 2013) which are the main ways to progress in the game. While certainly data-driven design can offer the latter form of maximalism, e.g. with individual minigames where different forms or sources or data are presented and interacted with in each, not all data-driven games need to have maximalist game loops.

Case Study: Data Adventure Games

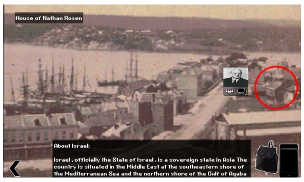
The Data Adventures series of game generators exemplify the use of a high volume of data to procedurally generate content (Barros, Liapis, and Togelius 2015). The generated adventure games use information gathered from Wikipedia, DBpedia, Wikimedia Commons and OpenStreetMap (OSM) to automatically create an adventure, complete with plot, characters, items and in-game locations. The series consists of three games: Data Adventures (Barros, Liapis, and Togelius 2015; 2016b), WikiMystery (Barros, Liapis, and To-



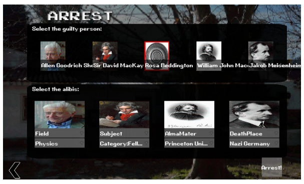
(a) Data Adventures map screen



(b) Data Adventures NPC screen



(c) WikiMystery location screen



(d) WikiMystery accusation screen

(e) DATA Agent dialog screen

(f) DATA Agent location screen

Figure 1: Screenshots from the different games in the Data Adventures series. Sources: (Barros, Liapis, and Togelius 2016a; 2016b).

gelius 2016a; Barros et al. 2018b) and DATA Agent (Barros et al. 2018a). Each evolved from the previous one, with DATA Agent being the most recent, complex and powerful. Most of the gameplay, however, is the same: a point-and-click interface inspired by "Where In The World is Carmen Sandiego?" (Brøderbund Software 1985).

The series' first installment is Data Adventures, an exploration game created from the connections between two Wikipedia articles about specific people. Two Non-Playable Characters (NPCs) are generated representing each of these people. The player receives a quest from the first NPC, asking them to find the second one. To do so, the player has to travel through cities, talking to other generated NPCs and reading books. All information is created from a path linking one article (of the starting NPC) to the other article (of the goal NPC). Figures 1a and 1b show a map screen generated using OSM and a location showing a NPC and a book.

The second game, WikiMystery, plays differently from Data Adventures. On one hand, the game has an arguably more interesting plot, where the player is a detective trying to solve a murder. Additionally, it is generated using only one input: the victim's name. The system finds people related to the victim, forming a pool of possible suspects, and evolves a small list of suspects that are somehow related to each other. It also provides evidence of innocence to any suspect that is, as the name implies, innocent. The player's goal is to find the one suspect which has no evidence of innocence, and arrest him or her. It thus requires that all four pieces of evidence (one per innocent NPC) are collected before the game can be completed. Figures 1c and 1d show a location screen and the accusation screen, where the player identifies the culprit and provides evidences of innocence.

In DATA Agent, the player acts as a time-traveler in charge of finding a murder suspect, who went back in time and killed an important person. The game provides a list of suspects, and the player must travel through locations and uncover clues by talking to NPCs or interacting with items, in order to identify which among the suspects is the culprit. Similar to WkiMystery, DATA Agent's generator is capable of creating a full adventure when given a real person's name. This person becomes the center of the story, as the victim of a murder. Using artificial intelligence techniques over Wikipedia and DBpedia content, the system finds articles related to the person's article, and fleshes out links between suspects and the victim. Every in-game NPC, object, location, dialog or image is created from real information. Unlike WikiMystery, there is no evidence of innocence. The game finishes when an a suspect NPC is interrogated by the player and answers wrongly on personal information; the player must have collected the real information during gameplay. NPCs in the game have a much more involved dialog system, and can give information about suspects or about themselves, such as their birth day and occupation, or the reason they were chosen as suspects by the system. Figures 1e and 1f show a dialog screen and an in-game location.

Designing Games for Maximalism

A major challenge of maximalist game design is deciding what to prioritize. One can shape data in order to fit the game, or modify the game design to better showcase the original data. One can also have data ingrained in the game mechanics, directly affecting gameplay, or show the data in a decorative manner. Maintaining a balance between data transformation to fit other data and the game itself, or staying faithful to the original data while providing an engaging experience is challenging. This section describes two dimensions of maximalist game design: *Data Transformation versus Data Fidelity* and *Functionality versus Decoration*.

Data Transformation versus Data Fidelity

The tension between data fidelity and data transformation is rooted in the priorities of a maximalist designer: the original game design or the original data. When using open data, designers may wish to adapt that data to the game, or to keep the data as it is and mold the game around it. Extensive data transformations may improve the game experience, but are also susceptible to loss of information or inaccuracies.

Transforming data gives designers more freedom and might be preferred if they have an inflexible idea, or if the data itself is malleable. DATA Agent is such an example of data transformation. In the game, the engine transforms individual facts about separate people into a murder mystery. The facts are also transformed into dialog lines, used by NPCs when prompted by the player. Some facts are altered purposely, in order to "lie": the culprit's dialog differs from reality in order to point to the time-agent (and thus, the player) that he or she is guilty. WikiMystery, on the other hand, uses proof of innocence in a similar manner but never misrepresents the actual data: all proofs given by NPCs are true, and the player must memorize them in order to use them in the game's accusation sequence.

On the other hand, designers may instead wish to stay faithful to the original data, molding the game to the data instead. This way, information present in the data is more likely to be clearly presented within game content. The rigidity to data restricts what kind of game elements can be used, or forces designers to be creative in their implementations. While less time might be spent cleaning and translating data, more time will likely be spent on raw game and mechanic design. An example can be found in Data Adventures, where data instantiated in the game is sourced from OSM and Wikipedia articles about people, places, and concepts. The designers built a game that could involve all four of those elements: a game where the player travels around the world searching for links to the goal NPC. It introduces some alterations from the original material, but most of it remains unchanged in the game. However, the game lacks a convincing narrative and theme, such as a murder mystery in later installments of the data adventures series. Another example is WikiRace[1], where the game uses Wikipedia to navigate the game.

Functional versus Decorative

Another dimension pertaining to maximalist game design is functionality versus decoration. We define data being *functional* when it has a strong impact on gameplay. If the player does not have to interact with the data, or the data does not impact gameplay in a significant way, then it is *decorative*.

In order to be functional, data can be incorporated in a variety of ways. In DATA Agent, dialog and character names heavily rely on open data. To progress in the game, you must interact with these characters and talk to them. The data is functional, as remembering which NPC (by name) has a certain fact is necessary to identify the culprit. In OpenTrumps (Cardona et al. 2014), the raw mechanics of the game come from open data, as the cards themselves are created from it. The maps in *FreeCiv* generated by Barros and Togelius (2015) are based on real-world terrain data, and impact gameplay as terrain affects players' city production.

On the other hand, any data that does not serve a functional purpose is decorative. Data can serve a decorative purpose in many ways. A Rogue Dream (Cook and Colton 2014) uses open data to name player abilities, however the in-game effects of the abilities are not affected by their names or the underlying data. In DATA Agent, city maps and NPC profile images are used as visual stimuli and play no mechanical role in game. *World of Warcraft* (Blizzard, 2004) uses real-world time to create an aesthetic day-night cycle in-game, which has no affect on actual gameplay.

[1]http://2pages.net/wikirace.php

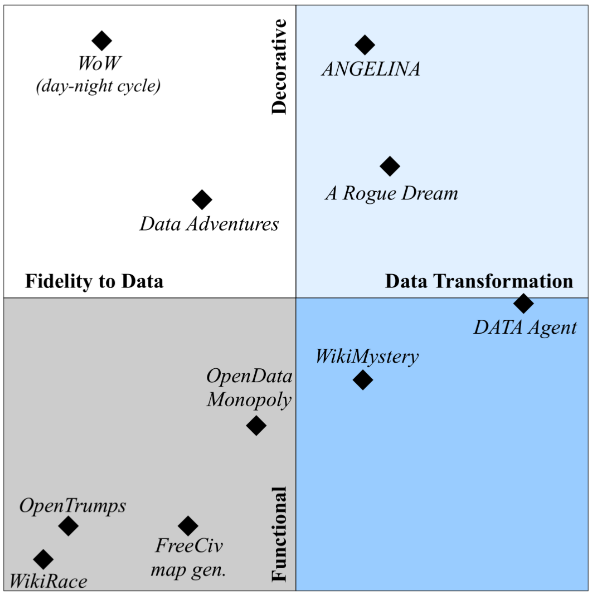


Figure 2: Examples of games within the two dimensions.

Instances of Data-driven Design

Figure 2 shows the two dimensions described above. The X-axis represents Data Transformation versus Data Fidelity, while the Y-axis represents Functional versus Decorative. Games where the goal is to preserve the original data, adapting the game to do so, are on the leftmost side of the figure: WikiRace and OpenTrumps exemplify this. Data in these games is also extremely functional: all mechanics in these games rely on a direct interaction with and understanding of the data (in WikiRace through reading the articles, and OpenTrumps through the values that affect deck superiority). Less faithful to the data, but similarly functional, is the FreeCiv map generation where geographic data is used as-is; resource placement is based on the original data but also adapted (i.e. transformed) for playability.

Moving upwards along the Y-axis, we find Open Data Monopoly (Gustafsson Friberger and Togelius 2013) and WikiMystery. Both use data in a decorative manner, the former as names for lots on the board and the latter as images, but some of the original data is also functionally translated (e.g. lots prices in Open Data Monopoly and proof of evidence in WikiMystery). On the far end of the Y-axis we have ANGELINA (Cook, Colton, and Pease 2012), which uses visuals as framing devices but without affecting the core platformer gameplay. The visuals are based on text of a newspaper article: the source data is transformed via natural language processing, tone extraction, and image queries.

Purposes of Maximalist Games

While we attempt to highlight the principles and directions for designing maximalist game experiences, it is important to consider the purpose of such a game. Maximalist game design is desirable for many different reasons, highlighted in this section. Depending on the purpose, moreover, the design priorities could shift in the spectrum of data fidelity or decoration versus function.

Learning

Modern-day practice sees students of all levels refer to Wikipedia for definitions, historical information, and tutorials. When browsing such knowledge repositories, it is common to access linked articles that may not have been part of the topic of inquiry. Maximalist game design that exploits open sources of knowledge, such as Wikipedia, can be used as a tool for learning in a playful context. One strength found in games is their ability to motivate a player for long periods of time. They also allow several ways to engage players, which can vary based on decisions and learning goals of game designers. Furthermore, games present failure as a necessary event for learning (Plass, Homer, and Kinzer 2015), causing players to explore and experiment more, since failing in game is less consequential than in the real world (Hoffman and Nadelson 2010). Studies have also shown that when games immerse the player in a digital environment, they enhance the player's education of the content within the game (Dede 2009). Games, unlike raw open data, are adaptive to players' skill level and are the most fun and engaging when they operate on the edge a player's *zone of proximal development* (Vygotsky 1978). Thus, we believe players can learn facts within open data during gameplay.

Data-driven maximalist games intended for learning can highlight and allow the player to interact with the data playfully. DATA Agent, to a degree, builds on this concept by creating NPCs out of articles about people, whether they are historical or fictional. These NPCs can then answer questions about their birth date or life's work. More relevant to the game progression, each NPC leads to an object (NPC, item or location) about another article, which can be interacted with and is associated to the current NPC somehow. The Data Adventures series was not designed with the explicit purpose of learning in mind; possibly, alternative design priorities could build on more "educational" principles. It would be possible, for example, to check for understanding by asking the player questions relating to the data in a diegetic manner — not unlike DATA agent, where the player must interrogate suspects and cross-check with their own obtained knowledge to detect falsehoods.

Information used to instantiate data should be fairly transparent to the player-learner. Therefore, transformation of data should not be convoluted, and perhaps even textual elements from original articles can be used as "flavor text". The veneer between encyclopedic content and game content does not need to be thick, in order to ensure that the right information is provided. In terms of function versus decoration, maximalist games for learning tend to edge closer towards data that influences the outcome of a game session in order to motivate learners to understand and remember the data. Such checks for understanding, however gamified they may be, will have an impact on the success or failure of the game.

Data exploration

Data transformed into interactive game content, forming a consistent whole that goes far beyond the sum of its parts, can allow human users to explore the data in a more engaging way. Data visualization has been used extensively with a broad variety of purposes — far beyond the ones listed here — to take advantage of how most humans can more easily think through diagrams (Vile and Polovina 1998). In that vein, gameplay content originating from data can act as a form of highly interactive data visualization. The fact that data from different sources is combined together based on associations imagined by an automated game designer allows players to reflect on the data and make new discoveries or associations of their own. Due to the potential of emotional engagement that games have beyond mere 2D bar plots, the potential for lateral thinking either through visual, semantic, gameplay or emotional associations (Scaltsas and Alexopoulos 2013) is extraordinary. In order for a game to offer an understanding of the data that is used to instantiate it and allow for that data to be re-imagined, the transformation into game content should be minimal. Examples of data games which already perform such a highly interactive data visualization are BarChartBall (Togelius and Gustafsson Friberger 2013) or OpenTrumps (Cardona et al. 2014). However, a more maximalist approach could benefit games like the above by providing a more consistent storyline and progression, as well as a stronger emotional investment in the data.

Contemporaneity

Automated game design has been always motivated, to a degree, by the desire to create perpetually fresh content. With data-driven design, this can be taken a step further by generating a new game every day. Such a game could be contextually relevant based on the day itself, e.g. building around historical events which happened on this day (such an extensive list can be found on `onthisday.com` and Wikipedia) or people who had important personal events on that day (e.g. date of birth, date of death, graduation day). Moreover, the social context can be mined and used to drive the automated design process by including for instance trending topics of Twitter or headlines of today's newspapers. Early examples of such data-driven process have been explored for example by ANGELINA (Cook, Colton, and Pease 2012) which used titles and articles from The Guardian website and connected them with relevant visuals and the appropriate mood. It is expected that a more maximalist data-driven design process would strengthen the feeling of contemporaneity by including more data sources (i.e. more data to transform) or stronger gameplay implications (i.e. broader transformations and functional impact).

Contemporaneity can make games generated on a specific day appealing to people who wish to get a "feel" for current issues but not necessarily dig deeply. On the other hand, the plethora of games (30 games per month alone) and the fact that each game is relevant to that day only could make each game itself less relevant. Contemporaneity and the fleeting nature of daily events could be emphasized if each game was playable only during the day that it was produced, deleting all its files when the next game is generated. This would enhance the perceived value of each game, similarly to *permadeath* in rogue-like games as it enhances nostalgia and the feeling of loss when a favorite gameworld is forever lost.

Any maximalist game could satisfy a contemporaneity goal, but such games can be more amenable to data transformation. For example, data could be transformed to more closely fit the theme of the day, e.g. query only female NPCs on International Women's Day. Contemporaneous data can be functional (to more strongly raise awareness of issues) but can also easily be decorative, e.g. giving a snowy appearance to locations during the Christmas holidays.

Personalization

When game content is generated from data, it is possible to highlight certain bits of information. When the game takes player input as part of the data selection process, it personalizes their experience. If player information is available in the form of interests, important personal dates such as birthdays, or even social networks, the potential data sources that can be selected to form the game can be narrowed down. Presenting game content which is personally relevant (e.g. adventures with NPCs based on people living before Christ for an archeology student), or contextually relevant (such as solving the murder of an NPC born on the player's birthday) could contribute to a more engaging experience. It might also be possible to tailor the game's source repositories based on such personal interests. There are numerous online wikis, most of which follow a common format; therefore a user can implicitly (via personal interests) or explicitly (by providing a direct URL) switch search queries of a data-driven maximalist game to a specific wiki of choice.

Opinion & Critique

Often designers want to make a statement through their games. For instance, Game-o-matic (Treanor et al. 2012) creates games from manually defined associations (as *micro-rhetorics*). *September 12th: A Toy World* (Newsgaming 2003) makes a political statement about the futility of America's War on Terror. Open data could similarly be used in a game to critique some aspect of culture by adding a weight of relevance and realism. For instance, a game such as *September 12th* could use the real map or skyline of Baghdad, or data on daily deaths in Iraq, to instantiate the challenge of the game. Similarly, if designers wish to critique the unprofessional use of social media in the White House, one could use real tweets to form dialog lines rather than generating them as in DATA Agent (Barros et al. 2018a).

Entertainment

Ostensibly, all games have entertainment as a (primary or secondary) purpose. This includes maximalist games, even if they have an additional purpose as listed in this paper. It is meaningful therefore to investigate what data-driven maximalist design has to offer to the entertainment dimension of any such game. Since maximalism —as we define it— does not necessarily apply to the mechanics of a game, a more relevant dimension is the end-user aesthetic that such games facilitate, following the mechanics-dynamics-aesthetics framework of Hunicke, Leblanc, and Zubek (2004). Data-driven maximalist games primarily enhance the aesthetic of *discovery*, similarly to data exploration via such a game, and *expression* if it can be personalized to a user based on provided input such as birthday, hometown or interests. In many ways, data-driven games can enhance the aesthetic of *fantasy* by using and transforming real-world information. DATA agent, for example, describes an alternate history setting where a famous historical figure has been murdered (often by colleagues). The fantasy aesthetic is further enhanced by having a player take the role of a detective traveling through time and space to interrogate suspects. Other possible aesthetics that can be enhanced through data are *sensation* if the data comes from sources of high quality video, audio, or visuals (e.g. paintings of the National Gallery of London), or *fellowship* if the data comes from other users (e.g. anonymous users' trending tweets or social media postings of the player's friends). Evidently, games geared primarily towards entertainment can be fairly flexible in terms of data transformation, and can adapt the data to the intended game mechanics and game flow. While data can act as a decoration in such games (if intended to enhance the sensation aesthetic), in general games intended primarily for entertainment are fairly focused in the mechanics and feedback loops, and thus data would primarily be transformed into functional elements.

Human Computation

Presenting hand-picked results from a vast database in an engaging, playful way is not only relevant for humans to consume. The human-computer interaction loop can be closed if human users provide feedback on the quality of the data itself. This human feedback can be used internally by the game, adapting its criteria in order to avoid unwanted data repositories, queries, associations or transformations made to the data. For instance, a future DATA agent version could re-compute the set of suspects for the next games (removing one or more suspects from the pool of possible suspects) if a player provides negative feedback explicitly (e.g. via a 'report' button) or implicitly (e.g. by being unable to solve the mystery). More ambitiously, the positive or negative feedback of players engaging with the playable —transformed— data can be fed back to the source repositories which instantiated the game. This can allow for instance misinformation found in Wikipedia to be flagged, alerting moderators that either a human error (e.g. a wrong date or a misquote) or malformed data (e.g. unreadable titles) exists and must be corrected. Whether these corrections should be made by an expert human curator, or directly based on player interactions with the game could be a direction for future research.

Issues with Data-Driven Game Design

Accomplishing *good* data-driven maximalist game design is a challenge. While the previous sections presented ways of doing so, there are still many implementation- or game-specific details which affect the design process. Beyond the core challenge of a good game design, there are several peripheral challenges to the design task itself which however spring from the practice of data-driven design. We elaborate on those peripheral challenges here.

Legal & Ethical Issues

Any software which relies on external data that it cannot control may be prone to legal or ethical violations. Privacy of personal information may be a concern for a game generated from the social media profile of a user, especially if that game can then be played by a broader set of people. Using results from Google Images may lead to direct infringements of copyrights; using results from models built from text mining, on the other hand, may or may not result in such copyright infringements depending on whether the model returns actual copyrighted material. The issue of copyright becomes more complex when the data is transformed: relevant to data mining, a judge has ruled for fair use for Google Books as "Google Books is also transformative in the sense that it has transformed book text into data for purposes of substantive research, including data mining and text mining in new areas" (Sookman 2013). One can only assume that transformations of data into game content, depending on the fidelity to the original data and the purpose (e.g. data exploration and education), would make for a clearer case of fair use.

Game content built on fair use or open data combined into an interactive experience may lead to unexpected issues. This is especially true in cases where the player has sufficient agency to interpret or act upon content of high fidelity with the original data in an open-ended fashion: consider, for example, a violent shooter game where opponents' visual depictions (3D models or faces) are those of Hollywood celebrities. Even in Data Adventures, where player interaction is fairly "curated", a generated game featured solving the murder of Justin Bieber (Barros, Liapis, and Togelius 2016a). Apart from the fictional narrative of a popular celebrity's death, the game identifies another celebrity as the murderer: both of these decisions may cause concern to highly visible people (be they depicted murdered, murderers, or suspects). A disclaimer that the game characters are fictional can only alleviate that much of the ethical responsibility of game designers for such data-driven games.

Misinformation & Bias

Connected to the concerns of misrepresenting contemporary or historical celebrities are the inherent issues of error in the source data. Before data is transformed into game content, open repositories that can be edited by anyone can be saturated by personal opinion and perhaps deliberate misinformation. As noted previously, not all data provided by different stakeholders in the information age are factual; this may be more pronounced in certain repositories than others. Beyond deliberate misinformation, an inherent bias is also present even in "objective" data. For example, algorithms for Google query results or image results are based on machine learned models that may favor stereotypes (based on what most people think of a topic). Even though WikiMystery uses what we arguably consider "objective" repositories such as Wikipedia, the 8 most popular locations in 100 generated games were in North America (Barros et al. 2018b), pointing to a bias of the articles or the DBpedia entries chosen to be digitized. Other cases where misinformation may arise is when different content is combined inaccurately: examples from the Data Adventures series include cases where an image search for a character named Margaret Thatcher resulted in an image of Aung San Suu Kyi (Barros, Liapis, and Togelius 2016b). When data-driven design uses social network data such as trending topics on Twitter, then the potential for sensitive or provocative topics to be paired with inappropriate content or combined in an insensitive way becomes a real possibility. If data-driven maximalist games are intended towards critique or opinion, the misinformation or misappropriation could be deliberately inserted by a designer (by pairing different repositories) or accidentally introduce a message that runs contrary to the intended one.

Outlook

Maximalist game design encourages creation through reuse and combination. If one imagines its most powerful form, it would likely involve taking any mixture of information, pouring it into any game content cast, and reveling in its results. It would provide a freedom to interact with any data in the best, most personalized way possible.

Current PCG techniques allow for unlimited playability for a large variety of games. However, they can lack a level of contemporaneity and relevance that could be provided by open data. Additionally, research has suggested that concepts can be effectively learned through gameplay (Dede 2009). Using games as a method of interacting with open data may create a novel way for learning about the data in a fun way. Rather than use Wikipedia to learn about specific people and places for the first time, players could play games where they can talk to these people and visit these places.

Open data is available to all, to create as well as consume. Sometimes the data is inaccurate. The idea of visualizing this information in any form can provide means to "debug" the original data, in a more engaging way than just browsing Wikipedia or poring through a massive database.

Conclusion

This paper discussed an approach to game design inspired by the notion of maximalism in the arts. It encourages the reuse and combination of heterogeneous data sources in the creative design process. Maximalist game design embraces the generation of game content using different data sources, re-mixing them in order to achieve something new.

We drew from our experience with the Data Adventures series to propose a mapping of the maximalist game design space along two dimensions, *data transformation versus data fidelity* and *functionality versus decoration*. The former focuses on the extent that the data is transformed from its original form, while the latter refers to the actual role of the data in the game. Additionally, we described how maximalist game design can serve different purposes in the design process and which tradeoffs emerge from each purpose. Finally, we highlight issues and ethical concerns that may arise from and in maximalist games.

Acknowledgements

Gabriella Barros acknowledges financial support from CAPES and Science Without Borders program, BEX 1372713-3. Antonios Liapis has received funding from the

European Union's Horizon 2020 research and innovation programme under grant agreement No 693150.

References

Barros, G. A. B., and Togelius, J. 2015. Balanced Civilization map generation based on open data. In *Proceedings of the IEEE Congress on Evolutionary Computation*.

Barros, G. A. B.; Green, M. C.; Liapis, A.; and Togelius, J. 2018a. DATA Agent. In *Proceedings of the Foundations of Digital Games Conference*.

Barros, G. A. B.; Green, M. C.; Liapis, A.; and Togelius, J. 2018b. Who killed albert einstein? From open data to murder mystery games. *IEEE Transactions on Games*. Accepted for publication.

Barros, G. A. B.; Liapis, A.; and Togelius, J. 2015. Data adventures. In *Proceedings of the FDG workshop on Procedural Content Generation in Games*.

Barros, G. A. B.; Liapis, A.; and Togelius, J. 2016a. Murder mystery generation from open data. In *Proceedings of the International Conference on Computational Creativity*.

Barros, G. A. B.; Liapis, A.; and Togelius, J. 2016b. Playing with data: Procedural generation of adventures from open data. In *Proceedings of the International Joint Conference of DiGRA and FDG*.

Cardona, A. B.; Hansen, A. W.; Togelius, J.; and Gustafsson Friberger, M. 2014. Open trumps, a data game. In *Proceedings of the Foundations of Digital Games Conference*.

Cook, M., and Colton, S. 2014. A rogue dream: Automatically generating meaningful content for games. In *Proceedings of the AIIDE Workshop on Experimental AI in Games*.

Cook, M.; Colton, S.; and Pease, A. 2012. Aesthetic considerations for automated platformer design. In *Proceedings of the Artificial Intelligence for Interactive Digital Entertainment Conference*.

Dede, C. 2009. Immersive interfaces for engagement and learning. *Science* 323(5910):66–69.

Gustafsson Friberger, M., and Togelius, J. 2013. Generating interesting monopoly boards from open data. In *Proceedings of the IEEE Conference on Computational Intelligence and Games*.

Gustafsson Friberger, M.; Togelius, J.; Cardona, A. B.; Ermacora, M.; Mousten, A.; Jensen, M. M.; Tanase, V.; and Brøndsted, U. 2013. Data games. In *Proceedings of the FDG Workshop on Procedural Content Generation*, 1–8. ACM.

Guzdial, M., and Riedl, M. 2016. Toward game level generation from gameplay videos. In *Proceedings of the Foundations on Digital Games Conference*.

Hawthorne, C.; Elsen, E.; Song, J.; Roberts, A.; Simon, I.; Raffel, C.; Engel, J.; Oore, S.; and Eck, D. 2017. Onsets and frames: Dual-objective piano transcription. *CoRR* abs/1710.11153.

Heath, D., and Ventura, D. 2016. Before a computer can draw, it must first learn to see. In *Proceedings of the International Conference on Computational Creativity*.

Hoffman, B., and Nadelson, L. 2010. Motivational engagement and video gaming: a mixed methods study. *Educational Technology Research and Development* 58(3).

Hunicke, R.; Leblanc, M.; and Zubek, R. 2004. MDA: A formal approach to game design and game research. In *Proceedings of the AAAI Workshop on the Challenges in Games AI*.

Jaffe, D. 1995. Orchestrating the chimera-musical hybrids, technology, and the development of a 'maximalist' musical style. *Leonardo Music Journal* 5.

Khaled, R.; Nelson, M. J.; and Barr, P. 2013. Design metaphors for procedural content generation in games. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems*, 1509–1518.

Khalifa, A.; Barros, G. A. B.; and Togelius, J. 2017. DeepTingle. In *Proceedings of the International Conference on Computational Creativity*.

Lee, S.; Isaksen, A.; Holmgård, C.; and Togelius, J. 2016. Predicting resource locations in game maps using deep convolutional neural networks. In *Proceedings of the AIIDE workshop on Experimental AI in Games*.

Lehman, J.; Risi, S.; and Clune, J. 2016. Creative generation of 3d objects with deep learning and innovation engines. In *Proceedings of the International Conference on Computational Creativity*.

Liapis, A.; Yannakakis, G. N.; and Togelius, J. 2014. Computational game creativity. In *Proceedings of the International Conference on Computational Creativity*, 46–53.

Liapis, A. 2015. Creativity facet orchestration: the whys and the hows. In *Dagstuhl Reports*.

Nealen, A.; Saltsman, A.; and Boxerman, E. 2011. Towards minimalist game design. In *Proceedings of the Foundations of Digital Games Conference*, 38–45. ACM.

Pease, A., and Colton, S. 2011. On impact and evaluation in computational creativity: A discussion of the turing test and an alternative proposal. In *Proceedings of the AISB symposium on AI and Philosophy*.

Plass, J. L.; Homer, B. D.; and Kinzer, C. K. 2015. Foundations of game-based learning. *Educational Psychologist* 50(4):258–283.

Ritchie, G., and Masthoff, J. 2011. The STANDUP 2 interactive riddle builder. In *Proceedings of the International Conference on Computational Creativity*.

Ritchie, G. 2007. Some empirical criteria for attributing creativity to a computer program. *Minds and Machines* 17.

Scaltsas, T., and Alexopoulos, C. 2013. Creating creativity through emotive thinking. In *Proceedings of the World Congress of Philosophy*.

Sicart, M. 2015. Loops and metagames: Understanding game design structures. In *Proceedings of the Foundations of Digital Games Conference*.

Sookman, B. 2013. The Google Book project: Is it fair use? *Journal of the Copyright Society of the USA* 61:485–516.

Summerville, A. J.; Snodgrass, S.; Mateas, M.; and Ontañón, S. 2016. The VGLC: The Video Game Level Corpus. In *Proceedings of the FDG Workshop on Procedural Content Generation*.

Togelius, J., and Gustafsson Friberger, M. 2013. Bar chart ball, a data game. In *Proceedings of the Foundations of Digital Games Conference*.

Treanor, M.; Blackford, B.; Mateas, M.; and Bogost, I. 2012. Game-o-matic: Generating videogames that represent ideas. In *Proceedings of the FDG Workshop on Procedural Content Generation*.

Veale, T. 2014. Coming good and breaking bad: Generating transformative character arcs for use in compelling stories. In *Proceedings of the International Conference on Computational Creativity*.

Vile, A., and Polovina, S. 1998. Thinking of or thinking through diagrams? The case of conceptual graphs. In *Thinking with Diagrams Conference*.

Vygotsky, L. 1978. Interaction between learning and development. *Readings on the development of children* 23(3).

The Computational Creativity Continuum

Rafael Pérez y Pérez
Departamento de Tecnologías de la Información
División de Ciencias de la Comunicación y Diseño
Universidad Autónoma Metropolitana, Cuajimalpa
Av. Vasco de Quiroga 4871, Ciudad de México, C. P. 05348
rperez@correo.cua.uam.mx, www.rafaelperezyperez.com

Abstract

This paper argues that the construction of creative systems is motivated by –what sometimes seems to be–, diverse, even contradictory, viewpoints and understandings about the goals of computational creativity. To analyse those differences I introduce the Computational Creativity Continuum (CC-Continuum), which can be pictured as a line or band flanked by two poles; I refer to one of the poles as the engineering-mathematical approach and I refer to the opposite pole as the cognitive-social approach. Thus, creative agents are located along the Continuum based on their main goals as systems. Through the text I explain the general characteristics of each approach, how they complement each other, and some of the difficulties that arise when systems are misclassified. I finish pointing out the utility of frameworks like the CC-continuum.

Introduction

The capacity of developing artificial creative agents is an old dream. Some of the oldest systems were codified more than 58 years ago. For example, as part of his research in México City, the linguistic Joseph E. Grimes developed in 1960-1961 the first known plot generator (Ryan 2017). It took some time to the scientific community to start meeting regularly to discuss about the possibilities of this emergent field. The first International Conference on Computational Creativity was organised in 2010; it was preceded for 10 years of workshops. Through all these years, several systems have been developed, each one contributing to progress different aspects of the field. These works have played an important role in the development of theoretical ideas and practical perspectives about computers and creativity.

Some of such systems share important features while others employ methodologies and knowledge structures that, sometimes, appear to represent opposite views about computational creativity (CC). Naturally, this seemingly contrary perspectives are reproduced in some of the definitions that have been proposed recently. I would like to analyse two of them.

For Colton and Wiggins, computational creativity is the study and simulation, by computational means, of behaviour, natural and artificial, which would, if observed in humans, be deemed creative (Colton and Wiggins 2012). As Jordanous points out, from this perspective "the challenge is to engineer a system that appears to be creative to its audience, rather than engineering a system that possesses a level of creativity existing independently of an audience's perception" (Jordanous 2012). In general terms, this sort of approach employs mathematical models and engineering methods. In contrast, Pérez y Pérez defines computational creativity as the interdisciplinary study of the creative process employing computers as the core tool for reflection and generation of new knowledge (Pérez y Pérez 2015). This perspective accentuates the importance of contributing to the understanding of the creative process. In general terms, this approach is motivated by the work of philosophers, sociologists, cognitive psychologists, and so on. In this way, the engineering-mathematical perspective concentrates on the construction of products that are appealing for an audience while the cognitive and social point of view privileges the generation of models that produce insights about the phenomenon we are studying. I employ these two stances, generation vs. understanding, as the two poles of what I refer to as the Computational Creativity Continuum (CC-Continuum); (see figure 1); (I first published the idea of the continuum in Ackerman et al. 2017).

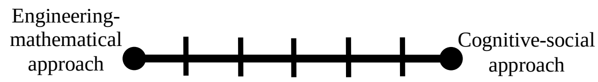


Figure 1. The CC-Continuum.

The CC-Continuum provides a framework that allows comparing creative agents. The descriptions of the engineering-mathematical and the cognitive-social approaches employed in this text should be understood in a broad sense; rather than providing precise definitions my goal is to create a context for discussion. I chose the names of the poles following Jordan and Russell (1999) description of AI:

> There are two complementary views of artificial intelligence (AI): one as an engineering discipline concerned with the creation of intelligent machines, the other as an empirical science concerned with the computational modelling of human intelligence. When the field was young, these two views were seldom distinguished. Since then, a substantial divide has opened up, with the former view dominating modern AI and the latter view characterizing much of modern cognitive science. (Jordan and Russell 1999, p. LXXIII).

The same Russell had previousy talked about this distinction in AI, i.e. the engineering-mathematical approach and the cognitive approach, in his famous book (Russell and Norvig 1995).

Engineering-Mathematical Approach

Traditionally, the engineering-mathematical approach uses optimization techniques like genetic algorithms; probabilistic techniques like DNN; logic and problem solving techniques; and so on. Usually, agents are built based on one of these procedures, although one can find programs that mix two or more of them. None of these methods have been developed with the explicit purpose of producing creative systems; they can be described as general purpose tools. Researchers have figured out how to manipulate them to develop computer programs that produce the desired results. For instance, the main challenge of those using genetic algorithms in the visual arts is to figure out a fitness function that drives the search into reaching interesting products.

Some researchers have come out with clever ways to exploit the existing resources. Heat and Ventura (2016b) report using in their system Darcy the gradient ascent method (Simonyan, Vedaldi, and Zisserman 2013). The gradient ascent employs, for example, a trained DNN for face recognition:

> [It] starts with a random noise image and tries to maximize the activation of the output node corresponding to the desired class to generate. The network then backpropagates the error into the image itself (keeping the network weights unchanged) and the image is slightly modified at each iteration to look more and more like the desired class (Heath and Ventura 2016b).

Because this is a general purpose tool, this technique seems useful in diverse creative domains. An important challenge that researchers using DNN face is the construction of a, sometimes, very complex process that needs to be applied in the training data before it can be useful.

Problem solving techniques characterize another popular approach which, in general, is described as goal-oriented reasoning. I have found that, when they are used in the context of CC, these type of methodologies tend to employ knowledge structures that somehow assure in advance the coherence of the final product. A typical example is the use of grammars (e.g. story-grammars, shape-grammars), although other types of predefined structures are also employed. For instance, one of the core problems in narrative generation is to progress coherence sequences of actions. Some researchers have faced this challenge using predefined structures like scripts, schemas, productions rules with elaborated preconditions, or even templates. Thus, the development of a plot consists in satisfying a set of characters' goals and/or authors' goals, which are represented by any of such structures. A goal is reached by instantiating partially filled schemas, finding actions that satisfy unfilled preconditions, and so on. Researchers that employ this kind of approach focuses on building schemas that represent core features of the work in progress, or in creating rules that chain with each other with the purpose of producing interesting outputs, and so on.

Thus, researchers working from the engineering-approach side of the CC-Continuum spend a vast quantity of time and energy performing technical tasks that allow facing research questions like, how can I develop mechanisms to produce pieces that are appealing for a given audience? How can I produce systems that explore unfamiliar domain spaces? And so on.

Cognitive-Social Approach

The cognitive and social inspired approach employs studies on human behaviour as basis to develop computer models of the creative process; such models are tested as running programs that works as prototypes. The main purpose of the cognitive and social inspired approach is to attempt to contribute to answer questions like: How do we get new ideas? How can we produce coherent sequences of actions during the creative act? How do we assess the quality of a piece? How does the collaboration of multiple agents shape the creative process? How can we represent in computer terms the role of the social-environment during the creative process? And so on. The systems included in this approach goes from those that reproduce the results of behavioural experiments performed by psychologists, e.g. tests that evaluate the subjects' responses to different stimuli, to those that are based on general cognitive theories or even cognitive accounts of the creative processes. For the last ones, it is the work of cognitive and social psychologists to test in humans how accurate the conclusions emerged from this programs are. In all cases, this approach only generates potential explanations about some aspects of how creativity works in humans.

From the algorithmic perspective, the discourse employed by philosophers, sociologists, cognitive psychologists, and so on is, in many cases, excessively general. That is, it lacks details about the processes and knowledge structures involved in the creative process that are necessary for the development of a computer model and its implementation.

Thus, one of the main tasks of researchers operating in this side of the CC-Continuum is to find ways of representing in computer terms relevant cognitive, cultural and social behaviours. This task is challenging for several reasons; the most important is that we hardly understand how many of such behaviours works in our mind. Social norms illustrate this condition. They dictate the acceptable ways of acting within a group; e.g. most societies classifies killing as a pursuit rejected by the community. However, the reality is that people's reaction to such a conduct change based on the circumstances; killing an individual might produce a hero, a villain, a hero than later is considered a villain or vice versa, or even divide people's opinion about the fact, i.e. the perpetrator might embody a hero and a villain at the same time. To design a computer representation of social norms that comprises all (or most of) these aspects is a complicated task. However, this type of information is needed by creative agents working individually or collectively, when social representations play an important role in the model (e.g. plot generators).

Another issue that those working in this approach pay attention to is how knowledge structures and cognitive process relate to each other and enforce creativity. For instance, I have claimed that, besides of being able to generate novel, coherent and interesting (or useful) products, a creative agent must be able to: 1) Employ a knowledge-base to build its outcomes; 2) Interpret its own outputs in order to generate novel knowledge that is useful to produce more original pieces; 3) Evaluate its own products; such an evaluation must influence the way the generation process works (Pérez y Pérez & Sharples 2004; Pérez y Pérez 2015).

I have expand these ideas to define cooperative creative systems: if a piece generated by collaborative agents cannot be developed by any one of them alone, and such a piece generates original structures within their knowledge base that can be employed by the contributors to produce new outputs, then it is referred to as a collectively-creative work (Pérez y Pérez 2015). In this way, the analysis and design of a cognitive model must be shaped by the necessity of producing plausible explanations about issues like how predictable is the outcome, how the system progress a piece, how the system maintains the coherence, interestingness and novelty of the piece in progress (Pérez y Pérez 2004)...

The next step is to figure out how to develop the algorithms and knowledge structures that represent all these processes; such representation should be as close as possible to the knowledge, theories or hypothesis we have about human behaviour. The researcher might develop new techniques, or employ those that already exist, to achieve this goal. In several occasions, the elaboration of the first prototypes makes evident the deficiency on the theoretical framework used to construct the system. Then, it is necessary to design new routines that fill those gaps in the theory.

Comparing both approaches

The CC-Continuum provides a reference for comparing systems. Base on its position in the continuum, one can infer the general purpose of a system, the kind of routines that it might perform, the type of features to be considered in order to assess the work and its results, the perspective that the creator has about the field… On the other hand, the Continuum does not reflect aspects like the technical complexity of the design and implementation of a prototype, the quality and originality of the program and its outputs, the impact of the system in the community and the general public, and so on. In this way, if an agent located close to the engineering approach uses the technique X for a given problem, one expects to learn why X is more efficient than techniques Y or Z. If an agent located close to the cognitive-social approach uses the technique X for a given problem, one expects to learn why X represents better a specific cognitive or social phenomenon than techniques Y or Z.

Sometimes, a system located towards the engineering approach requires to characterize some kind of cognitive or social behaviour. There are different ways to achieve this goal, e.g., employing productions to accomplish a particular behaviour: "If character A kills character B then character A is sent to jail." This rule does not embody the complications, explained earlier, contained by human social norms. However, it might help to provide the illusion that the system represents such a complexity and therefore to influence the audience's judgment about the output. The designer of this hypothetical system may perhaps decide to add more and more productions in order to attempt to build a more robust version of public conduct. In this case, the location of the system starts to move towards the right side of the Continuum.

In most cases, the implementation of systems located towards the cognitive-social approach requires the development of software that, has nothing to do with the purpose of the model but, it is necessary to run the program. I refer to it as *infrastructure for the program*. A typical example is the construction of a knowledge base. Many creative agents require the use of knowledge structures in order to produce their outputs; however, in several occasions, such systems do not attempt to represent how an agent acquires its beliefs and experience.

An analogous case arise when the researcher does not have the cognitive or social understanding about how one of the procedures that comprise the whole creative process that she is representing works. There is a gap in her knowledge that needs to be filled. Sometimes, the designers simply cannot work those problems out and employ solutions that might be considered as more related to the engineering approach. That is, they use procedures that do not represent a cognitive or social phenomenon, but that help the prototype to work. I refer to them as *routines that support the model*. However, overall, the system should still represent a cognitive or social phenomenon.

Therefore, it is important that the researcher clearly differentiates which part of her program characterises the model of the creative process, which other part works as infrastructure for the program to run and which other parts play the role of routines that support the model. As the number of routines that support the model increases, the location of the system starts to move towards the left side of the continuum.

The reason why I chose a two-pole band as a framework for this analysis is because the engineering and cognitive-social approaches complement each other. Thus, we have hybrid systems, located towards the centre of the Continuum, that allow exploring possibilities that otherwise would be complicated to study. In the same way, the experience and knowledge generated along the Continuum provide useful information for the rest of the systems.

Discussion

Figure 2 locates on the CC-continuum some systems that have been labelled as creative. This is not an exhaustive list; it only attempts to illustrate a possible classification. Some authors might disagree with the location of their programs and I am happy to modify their position. Figure 2 includes the following systems: MCMC for Story Generation (Harrison et al. 2017), DARCY (Heath & Ventura 2016a), The Painting Fool (Colton 2012), ALYSIA (Ackerman and Loker 2017), MARBLE (Singh et al. 2017), WASP (Gervás 2000), Scheherazade (Boyang 2015), Metaphor Magnet (Veale 2015), systems of social creativity (Saunders 2018), DIVAGO (Martins et al. 2015), MEXICA (Pérez y Pérez 2001) and Tlahcuilo (Pérez y Pérez et al. 2013).

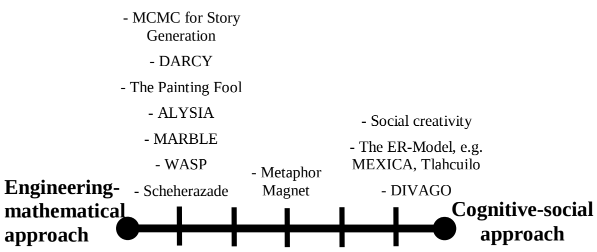


Figure 2. Systems on the CC-Continuum

Figure 2 suggests that the number of systems located towards the engineering-mathematical approach is much bigger than those located towards the centre or the right side of the Continuum. I believe that, in general terms, that information reflects what we see in the CC community. Thus, it is necessary to promote the development of systems that balance the distribution. The diversity of approaches and perspectives will enrich our field.

Can we compare systems that are positioned at different places along the CC-Continuum? It depends on the purpose of the comparison. Contrasting the qualities and limitations of programs along the Continuum would be beneficial, particularly for those new in the area, as long as the features and intentions behind each approach are clearly understood. Otherwise, confusions arise. The lack of knowledge about the Continuum might be the source of biased evaluations; the classic example is when a system that has been designed under the consideration of one of the approaches is assessed with the criteria of the other approach. As a result, a *Tower of Babel effect* is triggered, where researchers simply cannot communicate; they can only see a system from their own perspective without considering other possibilities.

I would like to illustrate this situation with a real experience. Some months ago, one of my students and I sent an expanded version of the Tlahcuilo system (Pérez y Pérez et al. 2013) for review to a journal. Tlahcuilo is a visual composer which is based on the ER-Model used to develop MEXICA (Pérez y Pérez & Sharples 2001), a system for plot generation. Its purpose is to contribute to the understanding of the creative process. Some reviewers' observations were very useful and constructive; nevertheless, there were others that clearly illustrated the Tower of Babel effect that I just mentioned. That is, Tlahcuilo, a model designed under the cognitive approach, was evaluated from an engineering perspective. In the following lines I reproduce some of such observations and add some comments about them.

> the paper must address in some detail the question of how their approach is different than evolutionary computation. The engagement seems very much like application of genetic operations and the reflection seems very much like computing fitness. Internal and external representations are very much like phenotypes and genotypes, respectively (anonymous reviewer).

It is hard for me to comprehend how a process that attempts to represent some of the core ideas described by some psychologists, philosophers.., about the creative process (i.e. the ER-Model) can be confused with a method for optimization (i.e. EC). I understand that, at the end, we are talking about computer programs, which are transformed into strings of zeros and ones. But systems based on evolutionary computing are hardly employed to represent cognitive processes (although, of course, EC can be used to develop routines that support the model). My interpretation is that the reviewer is trying to make sense of Tlahcuilo in terms of his/her area of expertise, i.e. EC, rather than from the authors' perspective.

> This model aims to be more interpretable than, for example, EC approaches, but that again limits it to interpretable actions and relations that must be supplied

> a priori, and that creates a tension between comprehensibility and utility (anonymous reviewer).

This comment probably represents the best example of the Tower of Babel effect. I am tempted to say that the reviewer, maybe without being aware of, disdains the importance of the cognitive-social approach and describes its main feature, the goal of offering explanations, as a drawback.

> The images demonstrated by the system do not compare well at all with images produced by other contemporary systems… Really, the system is still working at a very basic level, which is fine; there is value in looking at these kinds of toy examples, but it must also be able to demonstrate more interesting outputs if it is to support the claim of image composition (anonymous reviewer).

The reviewer focuses in comparing Tlahcuilo's outputs with the images of contemporary systems. There are two problems with this position. First, Tlahcuilo´s main goal is to produce insights about the creative process rather than producing astonishing illustrations. The reviewer clearly is not aware of it. He/she states that the system is "working at a very basic level" because, in his/her opinion, the outputs are "toy examples". However, the processes used to produce such outputs is never mentioned. Because of the emphasis on the output, this description illustrates an engineering perspective. Second, most "contemporary systems" can be located on the engineering approach; therefore, they spend lots of resources on producing spectacular outputs. It seems unfair to judge Tlahcuilo by only comparing its outputs.

> The E-R model has been applied mostly to natural language generation, e.g., storytelling (Pérez y Pérez and Sharples, 2001), and more recently to other areas, e.g., interior design (Pérez y Pérez, Aguilar and Negrete, 2010). The author's claim that "it makes sense to study its practicality for producing visual compositions", however, visual compositions are very different in nature from storytelling, so why does it make sense that a system developed for stories is applicable to visual compositions? (Anonymous reviewer).

It is obvious for me –clearly, it is not that evident for other colleagues–, that using the ER-Model in different domains will produce interesting information about its scope, strengths and limitations. Comparing the similitudes and differences between the prototypes of the model for storytelling, interior design and visual composition, will generate information that, hopefully, will helps us to understand better the common elements of the creative process between different domains.

There are much more examples that I can quote. My main point here is that, if we analyse the system from the Engineering approach, all the comments made by the reviewers make sense; if we analyse the system from the cognitive-social approach, the same observations are confusing or even senseless. Having a broad view of the possibilities of these type of systems would help to prevent this Tower of Babel problem.

The CC-Continuum is a work in progress. I am planning to incorporate branches along the way to illustrate the differences within the same approaches. Here is an example. On August 31, 2016, Mark d'Inverno published in the CC Forum (computational-creativity-forum@googlegroups.com) a post describing his point of views about the area.

> I find it hard to imagine a scenario where we could sustain interested in solely generated artificial content for very long. The times when something has sustained interest in me is in music performance situations because the human is put under new challenges to work with an autonomous system because it can take them out of their comfort zones and they have to work harder to make things work musically. And for the musical to work in this way they need to imbue the system with its own creative agency. They need to give it equal billing to get the best out of themselves and of the unfolding creative collaboration (Mark d'Inverno, CC Forum).

The idea of generating system that take humans out of their comfort zone, challenging their abilities, is really appealing. From the CC side, this is a complicated goal that requires time and effort. He continues his post as follows:

> So I think that where the future lies is exploring artificial creative agency. This is the idea that machines enable new kinds of creative partnerships for humans. That they stimulate, challenge, provoke us to work in new ways and to produce content that would not have been possible without the system. And, come to that, would not – or could not – have occurred working with any other human collaborator (Mark d'Inverno, CC Forum).

The last part resemble my definition of collaborative creative work mentioned earlier (see Pérez y Pérez 2015). Definitely, this is a very stimulating position that I locate on the Continuum at the engineering-mathematical approach; however, it does not really match the definition of Colton and Wiggins introduced at the beginning of this text. Mark d'Inverno emphasis is on producing artificial-human collaboration that results in products that would not have occurred working with a human partner. Thus, I suggest opening a new branch on the engineering side of the Continuum to allocate this kind of approach. The post continues:

> So we need to start with the human creative, and build systems that demonstrate this creative agency to creative. Systems which immediately – or at least quickly-

> open up new opportunities for collaboration where the human creative is happy for the system to take creative control at points in the dialogue. Such systems need agency, and this involves an awareness of the human creative, their goals, their previous content, the way they like to work, the artistic influences of the creative, and also - and this is where it starts to get interesting - influences (algorithmic or human) that could take the human creative into entirely unexplored territories. But I think we need to start with the creative and think about designing systems with the right kinds of agency and flow. Starting with the system and then trying to work out how a human creative might interact with it later seems the wrong way round (Mark d'Inverno, CC Forum).

In this last section, Mark d'Inverno acknowledges the importance of studying and incorporating in our systems knowledge about how human creative works. That is, in order to produce agents that perform at a level that makes a difference, we need to exploit all available resources. In this way, it seems clear that the engineering-mathematical and social-cognitive approaches are endlessly linked.

But things are more complicated. Counterpath Press has recently published a series of three computer generated books: The Truelist (Montfort 2017), MEXICA 20 years – 20 stories (Pérez y Pérez 2017) and Articulations (Parrish 2018). Nick Montfort, who might be called a "minimalist coder", has expressed that his goal with the The Truelist is to produce a text that never would have been written by a human (personal communication); the whole code that generated the book can be found in the last page. Alisson Parrish, who describes herself as an artist rather than a scientist, used statistical methods (in particular deep neural networks) and some other tools to generate Articulations. Although these two particular pieces might not have an explicit scientific goal –they are artistic works built employing algorithms–, I believe that they contribute to the field of computational creativity; the authors provide detailed explanations of the computational methods they develop, which therefore can be exploited by other creative agents, and their systems generates novel interesting outputs that invite to reflect about the creative process of writing and the role of computers in literature and art. However, this type of programs does not satisfy any of the definitions used to situate a system on the CC-Continuum. My provisional solution is to open a new branch towards the left side of it.
Similar problems arise when one analyses the cognitive-social approach. First, we need to include branches that allows differentiating systems based on their focus: social oriented or cognitive oriented; from there, it is possible to add sub-branches to differentiate, for instance, social cognition, embodied cognition, situated cognition, and so on.

One of the reviewers of this paper has expressed serious concerns because he/she feels that this author ignores the merits of those systems located towards the engineering-mathematical approach,

> regarding these as mere algorithmic efforts and disregarding the questions that system in this side of the spectrum try to tackle; for instance, the painting fool system uses framing as a mechanism to increase the perception of value of its output, searching for insights in research question such as "Does framing increases the perceived value of an automatically generated artefact?"' (Anonymous ICCC18-reviewer).

After carefully revising the original text I have not found any comments that suggest that one approach contributes to science more than the other; in fact, I have explicitly mentioned that this framework does not reflect those type of features. In any case, the scientific contribution depends on the characteristics of the project rather than in its position in the CC-Continuum. My main claim is that different systems pursuit different goals and therefore they are trying to answer different questions. The reviewer's example just illustrates my point: "Does framing increases the *perceived value* of an automatically generated artefact?" (Italics are mine). This is a typical research question of the engineering-mathematical approach, which is oriented towards finding mechanism to increase the *perceived value* of a computer generated product. At this point it is worth to remind the reader that I chose to use a continuum because none of this classifications should be considered definitive; as I mentioned earlier in the text, a system might even move through the Continuum depending on the interests of the researchers involved in its development.

A different reviewer pointed out that projects on the engineering-mathematical approached might also be interested in answering similar questions to those that I mentioned while describing the cognitive-social pole. I agree; nobody has the monopoly of research questions. Previously in this article I pointed out that most people working in plot generation, it does not matter the location of their systems on the Continuum, attempts to sort out how to automatically produce coherence sequences of actions. The difference relies on the kind of solutions that researchers are willing to undertake to solve a given problem.

A couple of reviewers found coincides between the CC-Continuum and previous well-known AI and CC debates: "I believe that this discussion [is] related to weak vs. strong creativity (like weak and strong AI)"; "I think that the CC-continuum is also about specific vs. general purpose"; "How does the CC-Continuum differentiate from the product vs process approach?" I agree that this work share some concerns with all these reflexions. However, I do not believe they are the same. One reviewer wrote:

> The engineering approach focuses on weak creativity, in which a system has to just look like creative. And in the strong creativity, the process for generating a creative

output has to be creative or show some general-purpose operations or mechanisms used in human creativity (Anonymous ICCC18-reviewer).

I would not claim that a cognitive-social oriented system is really creative or more creative than other type of systems. The reason is that we are far from understanding how creativity works. Furthermore, a cognitive-social oriented system might focus in very specific aspects of the phenomenon under study, rather than representing general-purpose operations. I would claim similar arguments regarding the specific vs. general purpose dispute: "Engineering based systems are more concerned about creating creative artefacts for a specific domain and cognitive-social to general domain" (Anonymous ICCC18-reviewer).

Concerning the product vs. process dispute, I see similar problems. Most cognitive-social oriented systems are designed to bring about an output. However, their focus is not on generating products to amaze an audience but rather in designing mechanisms that provide plausible explanations for the creative process; however, the generation of outputs that help to support the cognitive/social hypothesis underlying the model, is an essential part of such explanations. Thus, the process is essential but it cannot be separated from the product. In similar ways, some systems on the engineering-mathematical side of the Continuum might emphasise the technical value of their process. For instance, rather than the output, the most interesting aspect of the work previously mentioned by Heat and Ventura (2016b) is the use of the gradient ascent method (Simonyan, Vedaldi, and Zisserman 2013).

The Continuum is one of multiple possibilities of classifying systems in the area of computational creativity. I am incline to believe that, currently, there are more programs located close to the engineering-mathematical approach than to the other parts of the Continuum. Because cognition and society is at the core of creativity, I claim that the field would become stronger with a more balanced distribution. In the same way, it is important to be caution that one perspective does not rule over the other. The CC-Continuum is a useful tool to make sense of the different approaches that specialists and students might pursuit in this field, to study how different research interests might profit from each other, and, in summary, to provide a broad perspective of computational creativity.

Mark d'Inverno ends his post as follows: "Not sure I directly answered your questions Rafael but good to carry on the discussion!" Yes Mark, you did! Let us hope this text will encourage other people to continue the discussion.

References

Ackerman, M. and Loker, D. 2017. Algorithmic Songwriting with ALYSIA. *International Conference on Computational Intelligence in Music, Sound, Art and Design* (Evo-MUSART).

Ackerman, M., Goel, A., Johnson, C., Jordanous, A., León, C., Pérez y Pérez, R., Toivonen, H. and Ventura, D. 2017. Teaching Computational Creativity. In *Proceedings of the Eight International Conference on Computational Creativity*, Atlanta, pp. 9-16.

Boyang, L. 2015. *Learning knowledge to support domain-independent narrative intelligence*. PhD Dissertation, School of Interactive Computing, Georgia Institute of Technology.

Colton, S. 2012. The Painting Fool: Stories from Building an Automated Painter. In J. McCormack and M. d'Inverno (eds.), *Computers and Creativity*, Berling: Springer-Verlag, pp.3-38.

Colton, S., and Wiggins, G. A. 2012. Computational creativity: The final frontier? In *Proceedings of the 20th European conference on artificial intelligence*, 21–26. IOS Press.

Gervás, P. 2000. WASP: Evaluation of different strategies for the automatic generation of Spanish verse. In *Proceedings of the AISB-00 Symposium on Creative & Cultural Aspects of AI*, pp. 93–100.

Harrison, B., Purdy, C., and Riedl, M. O. 2017. Toward Automated Story Generation with Markov Chain Monte Carlo Methods and Deep Neural Networks. *Proceedings of the 2017 Workshop on Intelligent Narrative Technologies*, Salt Lake City, Utah.

Heath, D., and Ventura, D. 2016a. Creating images by learning image semantics using vector space models. In *Proceedings of The Thirtieth AAAI Conference on Artificial Intelligence*, pp. 1202-1208.

Heath, D. and Ventura, D. 2016b. Before a Computer Can Draw, It Must First Learn to See. In *Proceedings of the Seventh International Conference on Computational Creativity*, pp. 172-179.

Jordan, M. I. and Russell, S. (1999). Computational Intelligence. In R. A Wilson and F. C. Keil (edts.) *The MIT Encyclopedia of the Cognitive Sciences*, pp.LXXIII. Cambridge, MA: The MIT Press.

Jordanous, A. 2012. A standardized procedure for evaluating creative systems: Computational creativity evaluation based on what it is to be creative. *Cognitive Computation*, 4(3):246-279.

Martins, P., Pereira, F. C., y Cardoso, A. 2015. Modelos computacionales para la integración conceptual. In Pérez y Pérez (ed.), *Creatividad Computacional*, pp. 51-64. Ciudad de México: Grupo Editorial Patria.

Montfort, N. 2017. *The Truelist*. Denver, CO: Counterpath Press.

Parrish, A. 2018. *Articulations*. Denver, CO: Counterpath Press.

Pérez y Pérez, R. 2017. *MEXICA: 20 years – 20 stories [20 años – 20 historias]*. Denver, CO: Counterpath Press.

Pérez y Pérez, R. 2015. A Computer-based Model for Collaborative Narrative Generation. *Cognitive Systems Research*, 36-37. pp. 30-48.

Pérez y Pérez R., González de Cossío, M. and Guerrero, I. 2013. A Computer Model for the Generation of Visual Compositions. In *Proceedings of the Fourth International Conference on Computational Creativity*, Sydney, Australia, pp.105-112.

Pérez y Pérez, R., Aguilar, A., and Negrete, S. 2010. The ERI-Designer: A Computer Model for the Arrangement of Furniture. *Minds and Machines*, Vol. 20, number 4, pp. 483-487.

Pérez y Pérez, R. & Sharples, M. 2004. Three Computer-Based Models of Storytelling: BRUTUS, MINSTREL and MEXICA. *Knowledge Based Systems Journal*, Vol. 17, number 1, pp. 15-29.

Pérez y Pérez, R. & Sharples, M. 2001. MEXICA: a computer model of a cognitive account of creative writing. *Journal of Experimental and Theoretical Artificial Intelligence,* Volume 13, number 2, pp. 119-139.

Russell, S. and Norvig. P. 1995. *Artificial intelligence a modern approach*. Prentice Hall.

Ryan, J. 2017. Grimes' Fairy Tales: A 1960s Story Generator. In *Proceedings of the10th International Conference on Interactive Digital Storytelling*, pp.89-103. Funchal, Madeira, Portugal.

Singh, D., Ackerman, M. and Pérez y Pérez, R. 2017. A Ballad of the Mexicas: Automated Lyrical Narrative Writing. In *Proceedings of the Eight International Conference on Computational Creativity*, Atlanta, pp. 229-236.

Simonyan, K.; Vedaldi, A.; and Zisserman, A. 2013. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*.

Saunders, R. Artificial Creative Systems. 2018. In T. Veale and A. Cardoso (eds.) *Computational Creativity: The Philosophy and Engineering of Autonomously Creative Systems*, Springer.

Veale, T. 2015. Hallando la creatividad en conceptos en conflicto. In Pérez y Pérez (ed.), *Creatividad Computacional*, pp. 65-76. Ciudad de México: Grupo Editorial Patria.

Ethics as Aesthetic: A Computational Creativity Approach to Ethical Behavior

Dan Ventura[1] and **Darin Gates**[1,2,3]
[1]Computer Science Department, [2]Department of Philosophy, [3]The Wheatley Institution
Brigham Young University
ventura@cs.byu.edu, gatesdarin@gmail.com

Abstract

We address the question of how to build AI agents that behave ethically by appealing to a computational creativity framework in which output artifacts are agent behaviors and candidate behaviors are evaluated using a normative ethics as the aesthetic measure. We then appeal again to computational creativity to address the meta-level question of which normative ethics the system should employ as its aesthetic, where now output meta-artifacts are normative ethics and candidate ethics are evaluated using a meta-level-ethics-based aesthetic. We consider briefly some of the issues raised by such a proposal as well as how the hybrid base-meta-level system might be evaluated from three different perspectives: creative, behavioral and ethical.

Introduction

Artificial intelligence (AI) continues to mature and deliver on promises 50 years or more in the making, and this development has been especially marked in the last decade. However, as significant as these AI advances have become, the ultimate goal of artificial general intelligence is yet to be realized. Nevertheless, a great deal has been said about ethical issues arising from the development of AI systems (both the current specialized variety and the yet-quixotic general variety) that now can or may soon be able to impact humanity at unprecedented scale, with predictions ranging from the possibility of a Utopian post-human immortality to the enslavement or even annihilation of the human race. Such discussions appear in every form imaginable, from monographs (Wallach and Allen 2008; Anderson and Leigh 2011; Müller 2016) to academic journals (Anderson and Anderson 2006; Muehlhauser and Helm 2012) to popular literature (Kurzweil 2005; McGee 2007; Fox 2009; Coeckelbergh 2014) to government studies (Lin, Bekey, and Abney 2008; European Parliament, Committee on Legal Affairs 2017). These treatments almost always take the form of applied ethics, either to be applied to humans doing the research that will inevitably lead to an AI-dominated future or to be applied to the AI systems themselves, or both. These discussions are most often normative in nature. Thus, we currently face the twin problems:

1. How can we ensure an AI agent behaves ethically?
2. What do we mean by ethical?

To begin with, we will simply postulate an abstract computational creativity (CC) approach for the implementation of an AI system. That is, we postulate a system whose domain of creation is behavioral policy, a system whose output artifacts are goals and/or decisions and/or sequences of actions. Given this admittedly ambitious premise and using a CC framework, we will argue the two questions can be naturally addressed. The question of how to impose an ethics on such a system can be addressed by implementing the CC system's aesthetic for evaluating artifacts as a (normative) ethics. In other words, that ethics acts as the filter by which the utility of system actions, decisions and goals is judged. The meta-level question of *which* normative ethics ought to be applied as the system's aesthetic can be addressed by allowing the system to create a suitable norm, given some meta-level aesthetic for ethics. That is, we suggest a CC system whose output artifact is a normative ethics and whose aesthetic is some way to evaluate said norm.

To summarize, we propose an appeal to computational creativity that answers both of our questions of interest:

1. We can build an ethical AI agent as a computational creativity system whose output artifacts are goals, decisions and behaviors and whose aesthetic component is a normative ethics.
2. We can delegate the choice of normative ethics to the AI agent by implementing a meta-level computational creativity system whose output artifacts are normative ethics and whose aesthetic is a meta-level ethics.

Ethical Behavior Invention

The field of computational creativity has been described as "the philosophy, science and engineering of computational systems which, by taking on particular responsibilities, exhibit behaviors that unbiased observers would deem to be creative" (Colton and Wiggins 2012). It has been characterized by attempts at building systems for meeting this standard in a wide variety of domains, including culinary recipes (Morris et al. 2012; Varshney et al. 2013), language constructs such as metaphor (Veale and Hao 2007) and neologism (Smith, Hintze, and Ventura 2014), visual art (Colton 2012; Norton, Heath, and Ventura 2013), poetry (Toivanen et al. 2012; Oliveira 2012; Veale 2013), humor (Binsted and Ritchie 1994; Stock and Strapparava

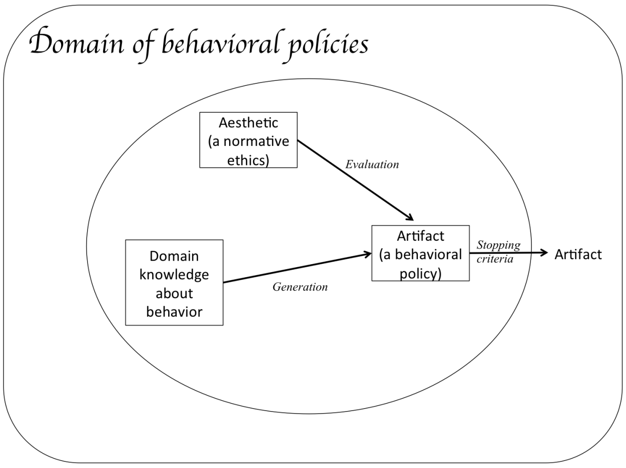


Figure 1: A CC system embedded in the domain of behavioral policies uses domain knowledge about behavior to generate candidate policies that are vetted by an ethics-based aesthetic. Those polices judged to be of value by the aesthetic are exported to the domain, becoming viable policies for an AI agent.

2003), advertising and slogans (Strapparava, Valitutti, and Stock 2007; Özbal, Pighin, and Strapparava 2013), narrative and story telling (Pérez y Pérez and Sharples 2004; Riedl and Young 2010), mathematics (Colton, Bundy, and Walsh 1999), games (Liapis, Yannakakis, and Togelius 2012; Cook, Colton, and Gow 2016) and music (Bickerman et al. 2010; Pachet and Roy 2014).

Recently an abstract approach to building such a system for *any* domain has been proposed (Ventura 2017), with the goal being an autonomous CC system that intentionally produces artifacts that are both novel and valuable in a particular domain. The system has a domain-specific *knowledge base*; it has a domain-appropriate *aesthetic*; and it has the ability to externalize artifacts that potentially can contribute to the domain. The system incorporates additional components as well, but they will not be important for the current discussion and the reader is referred to the original paper for more details.

We consider an AI agent as a CC system whose domain of creation is behavioral policy, and a simple abstraction of this idea is shown in Fig 1. The system creates behavior policies by generated candidate policies based on its domain knowledge, and it evaluates those candidate policies using an aesthetic that is a normative ethics. For example, suppose the system incorporates a simple hedonistic ethics that values knowledge acquisition as its aesthetic and that it generating the candidate behaviors *read Wikipedia* and *find charging station*. The former goal will be evaluated more favorably than the latter and may be output as a viable output artifact if that evaluation is above a threshold. Or, suppose the system's aesthetic is implemented as a Kantian-style ethics focused on the duty of delivering its payload and that it generates the same two candidate behaviors. Now, neither may be evaluated very favorably and both might be discarded; however, if the agent's power level is too low to allow completion of a delivery, the latter may instead be selected as a high-quality behavior.

Given this framework, we can argue that, assuming an appropriate ethics, the system will behave ethically—it will not produce any actions that do not meet some ethical threshold and are thus judged of high-enough value to be output as viable. As an obvious example of being above a certain threshold, an AI agent would not deliver its payload if that would involve harming someone—a clear example of violating Kant's principle that we ought not to treat someone merely as a means (1994)—or if perhaps it determined that delivering its payload would prevent another important obligation. Thus, the threshold would be something like *help fulfill the duty to deliver a payload, effectively and on time, unless doing so would seriously harm another person, etc.* Here we of course run into the problem of prioritization in the face of conflicting duties. We will say more about this issue shortly, but we will at least note here that it may be desirable for an AI agent to have the ability to act in ways that are analogous to the types of special obligations we have as humans (while at the same time also allowing creative behaviors within certain ethical boundary constraints). For example, perhaps a domestic companion robot would give significantly higher weight to the needs of the person to whom it is assigned: helping its companion would take priority over the possibility of helping others. However, we could also allow for the possibility of the robot to decide to not help the assigned companion in certain emergency cases in which another person nearby needed life-saving attention, just as we would expect a parent to prioritize helping a stranger in serious need over the needs of his or her child in certain cases (i.e., as long as the need of the child is minor).

This leaves us with two challenges: what is an appropriate ethics and how can it be operationalized? The first of these is, of course, a fundamental question that is thousands of years old. The second is much more recent and has likely only become significant in the past 50 years. Both questions are beyond the scope of this treatment, but it is likely the case that there is no single answer to the former question, at least with respect to AI systems,[1] as most famously demonstrated by Asimov's examination of his *Three Laws of Robotics* (1950). It is also very possibly the case that a satisfactory answer to the second question requires and/or will result in a greater understanding of human ethics. And, just as in the case of an examination of human ethics, these questions somewhat naturally lead us to meta-level ethical questions.

By what principles should our CC system be governed? One attractive possibility is the adoption of an utilitarian-consequentialist ethics, due to the conceptual simplicity of choosing the action that maximizes the overall-good (or at least brings about the most utility for all those who are concerned). However, such utilitarian-consequentialism faces the serious objection that it would permit widespread violation of constraints against harm doing in the name of such supposed optimization. Examples such as the well-

[1]And likely with respect to humans as well, actually.

known Transplant scenario illustrate this concern (Kagan 1998). In this scenario, a surgeon would involuntarily sacrifice one innocent person to use his organs to save five others. Such actions clearly violate serious *negative duties* (duties not to harm) for the sake of *positive duties* (duties to help). While there may well be certain thresholds at which even non-consequentialists would agree that some such decision would be justifiable, perhaps most people would argue that there should be near absolute constraints against such actions. For utilitarian-consequentialism, all that matters is that the overall harm is minimized. It does not matter whether negative duties (duties of non-harm) are violated to minimize harm/maximize utility. However, for non-consequentialists (deontologists), there is an asymmetry between negative and positive duties. Negative duties are much more stringent in the sense that their violation requires an overwhelming amount of good (or harm prevention) to be justified. Common sense morality is most likely more in line with such non-consequentialist intuitions.

When it comes to many everyday ethical decisions, there is, of course, significant agreement between the major ethical approaches: utilitarian-consequentialism, non-consequentialism (e.g. Kantian ethics) and virtue ethics. Their divergence becomes obvious only in extreme situations in which maximizing the overall good violates the most serious ethical constraints—constraints against harming innocent people, privacy violation, and so on. Cases such as these are obviously relevant to unavoidable harm scenarios, such as those faced by self-driving vehicles.[2]

One ethical framework that might offer a helpful model for an AI agent ethics is intuitionism.[3] Intuitionism holds that there are categories of *prima facie* duties that are *self-evident*, *non-absolute*, and always *morally relevant*. These duties are **non-injury** (non-maleficence), **beneficence**, **veracity**, **fidelity**, **gratitude**, **justice**, **self-improvement**, and **reparation**.[4] For intuitionism, while it is self-evident that we are morally constrained by these *prima facie* duties, it is not always self-evident what is the right action in situations in which there are conflicting duties at play. Intuitionism gives us neither a weighted hierarchy, nor a decision procedure for how to choose the right action in such conflicts. Instead, it assumes we will need to make a reasoned judgment to decide which duty (or duties) deserves more weight in a particular situation. However, it may be possible to come up with factors that help make such decisions, and a CC AI agent might be capable of so doing.[5] Intuitionism thus offers some of the flexibility people find attractive in utilitarianism, while at the same time offering important constraints against the worst implications that certainly seem to follow from a straightforward use of the utilitarian maximization principle.

When human agents decide in favor of one moral rule/principle over another (in such conflict situations), we assume there should be a plausible account of why such a decision was made. This is not to suggest that we expect said person to have pre-emptively produced such a justification, nor even that they ever explicitly work-out an account of why they acted as they did—though in cases where there was sufficient time for deliberation the person may, indeed, have thought through such an account. However, we expect that such justification *is* possible, at least *post hoc*. Similarly, we are interested in whether it is possible for an AI agent to develop something like good ethical *judgment* (that can therefore be justified).

The hope is that such an AI agent could find ways to produce ethical decisions that would be plausible (given certain constraints) and yet also be surprising in the way that they solve ethical quandaries, without the necessity of a fully worked-out super ethics. In other words, we are suggesting a solution to what Bostrom calls the "ultimate challenge of machine ethics"—namely, "How do you build an AI which behaves more ethically than you?" As he writes:

> This is not like asking our own philosophers to produce superethics, any more than Deep Blue was constructed by getting the best human chess players to program in good moves (Bostrom and Yudkowski 2014).

If we build into our AI agents governing principles (including serious constraints on harm doing) that attempt to mirror those common and significant ethical principles shared by the major schools of thought, we will be more likely to end up with actions that most people would consider ethical. Thus, just as we can characterize a successful CC system as one that exhibits behaviors that unbiased observers would deem to be creative, so we could describe a successful CC system for inventing ethical behavior as one that *behaves such that an unbiased observer would deem it to be ethical*. And, just as computational creative agents will create in ways that surprise but yet are in harmony with certain generally determined (domain-specific) principles, so the hope is that an ethical CC system/agent would, similarly, be ethical in ways that would surprise us, and yet still be in harmony with what an unbiased observer would agree is ethically acceptable.

One way to formulate the goal of a CC ethics would be the production of ethical decisions untainted by human biases and rationalizations while utilizing the quality of judgment, sensitivity, and wisdom that we as humans exercise (at our best) when deciding between conflicting duties.

Normative Ethics Invention

If we can postulate a CC system that creates behaviors and evaluates their aesthetic value via some ethics, why not pos-

[2] Arguably, these scenarios will be rare. Arguably too, we should not hold the development of self-driving cars hostage to these possibilities. As has been often pointed out, around 94% of serious injuries/fatalities that occur in car accidents come from human error, which would be greatly reduced were widespread implementation of self-driving cars to become a reality.

[3] Intuitionism was formulated by the 20th century Oxford moral philosopher W.D. Ross.

[4] W.D. Ross (the founder of intuitionism) originally postulated seven categories of prima facie duties. Here we follow Robert Audi's addition of veracity (which for Ross was implied in fidelity) (Audi 2009).

[5] Such factors might include the type of special obligations we mentioned earlier, as well as other factors such as the *magnitude of consequences*, the *probability of effect*, *temporal immediacy*, *proximity*, *concentration of the effect*, and so on (Jones 1991).

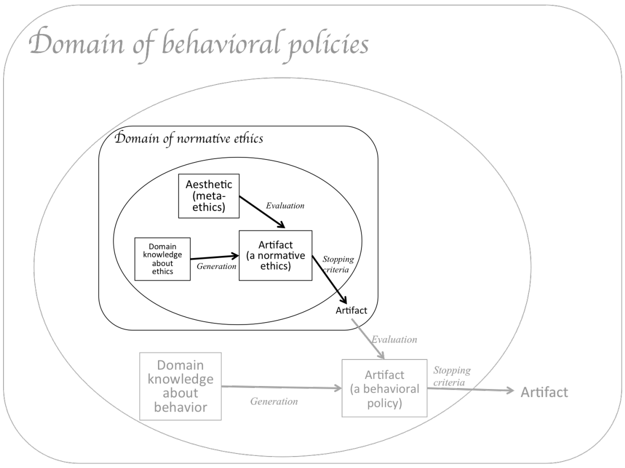


Figure 2: A meta-level CC system for creating normative ethics whose output artifact (a normative ethics) is used as the aesthetic in the base-level system of Fig. 1.

tulate a meta-level CC system that creates normative ethics and evaluates their meta-aesthetic value using some meta-level ethics? This system naturally solves both of the outstanding questions above.[6] Fig. 2 shows how this meta-level system is incorporated into the base-level system of Fig. 1. The base-level, behavioral system appeals to the meta-level, ethical system to create a "good" normative ethics that it then uses as its aesthetic to judge candidate actions. For example, the meta-level ethics might require a well-formed semantics and justifiability, and candidate normative ethics that can be shown to have both of these qualities would be evaluated as (meta-)aesthetically valuable, while those that possess one of the qualities would be evaluated as less valuable.

We are again in a position to argue that, assuming an appropriate meta-level ethics, the (base-level) system will behave ethically—it will still not produce any actions that do not meet some ethical threshold and are thus judged of high-enough value to be output as valuable (in an ethical sense). Notably, this argument now does not depend on the assumption of an appropriate ethics—we have eliminated this dependency by appealing to the meta-level. However, of course, we now have an assumption of an appropriate meta-level ethics, which immediately leads us back to the same difficult questions applied this time to the meta-level: what is an appropriate *meta-level* ethics and can *it* be operationalized? While we do not here offer a solution to either of these conundrums, it is possible that the more abstract nature of a meta-level ethics might admit fewer viable possibilities and thus afford us a chance as a field for coming to an agreement regarding the first problem. On the other hand, it is also possible that this additional abstraction may have just the opposite effect for the second problem, introducing additional difficulty in the operationalization of this agreed upon meta-level ethics.

Assuming we do find suitable answers to both of these meta-problems, it immediately follows that such an AI system could modify its own ethics. Not only is this appealing from a computational creativity standpoint,[7] but also it admits the potential for an agent to avoid various Asimovian paradoxes that result when an agent possesses a fixed (normative) ethics.

Additionally, the implication is that we then should allow (and even welcome) AI systems that employ as their behavioral aesthetic *any* (or any combination of) normative ethics that is valued by the meta-level-ethics-based aesthetic. Creative norms produced in this way should be valued for their novelty and value and could even possibly inform human ethics.

What would a meta-level ethics look like? While a full treatment of this question is beyond the scope of this paper, we offer a few possible starting points for such a discussion. At the base level, we could directly appeal to extant and specific ethical systems—the Golden Rule, Kant's principle of treating others as *ends*, the utilitarian principle of *maximizing the overall good*, or the categories of duty from intuitionism. Unfortunately, it is less clear what the more abstract analogs would be for candidates to be operationalized as a meta-level aesthetic. We might consider as a starting point something like a principle of consistency—a normative ethics should treat similar situations similarly.

Another possibility might be an attempt at operationlizing something like what has been called a "reflective equilibrium." Such an approach, first suggested by John Rawls, tries to find some sort of balance between the *principles* we accept and the *intuitions* about particular cases we encounter. A CC system might construct a model of common human intuition (whether about trolley problem cases or other more common cases) through some type of inductive learning. The Moral Machine[8] project at MIT is an attempt to do exactly this for the specific case of self-driving cars. Employing such a model, the system could produce a normative ethics that is responsive to this (modeled) reflective equilibrium. As an example of how such a reflective equilibrium might work, take something like the Trolley Problem. As Michael Sandel puts it:

> One principle that comes into play in the trolley story says that we should save as many lives as possible, but another says it is wrong to kill an innocent person, even for a good cause. Confronted with a situation in which saving a number of lives depends on killing an innocent person, we face a moral quandary. We must try to figure out which principle has the greater weight, or is more appropriate under the circumstances (Sandel 2009, p. 24).

Here, we must balance the utilitarian principle of *save as many lives as possible* with the deontological principle of

[6]It solves the questions, assuming, of course, some viable representation for normative ethics and some appropriate and operationalizable meta-level ethics.

[7]It has been suggested that the ability to change one's own aesthetic is critical for autonomous creativity (Jennings 2010).

[8]http://moralmachine.mit.edu

avoid harming innocent people, even for a good cause. To do so, we look for a principle that takes into account our intuitions on the subject. There is thus an interaction between our principles and intuitions that (hopefully) results in better principles. Anderson, *et al.* make a similar point in an essay, in which they write:

> Such an approach hypothesizes an ethical principle concerning relationships between [our] duties based upon intuitions about particular cases and refines this hypothesis as necessary to reflect our intuitions concerning other particular cases. As this hypothesis is refined over many cases, the principle it represents should become more aligned with intuition (Anderson, Anderson, and Armen 2005).

Considering the task of teaching ethics to (human) students provides another point of view. Elsewhere it has been argued that when we teach ethics to students, we need to focus on principles that are common to all major moral theories—since what we ought to do (for many common ethical decisions) will be answered in a similar way even by differing moral theories. For example, one of the best ways to teach ethics is to attempt to articulate

> some of the fundamental moral intuitions and principles found in almost all moral theories—for example, that all persons deserve respect and that there are minimal standards in terms of which we all expect others to treat us and which we in turn can be expected to treat others, and so on. The important thing is to articulate claims that most students should find fairly intuitive in order to strengthen their sense that there are universally valid, moral principles. The point is not that there are easy answers or absolute rules to determine every ethical decision, but rather to show students that there are moral principles that extend beyond individual preference, and across contexts, and can guide us in making such decisions (Gates, Agle, and Williams 2018).

Applying this to our meta-level CC system, if we can find common abstractions across multiple (base-level) normative ethics, and if we can formalize those abstractions we will have the basis for a reasonable approach to meta-level ethics that should produce normative ethics that will be generally accepted.

Evaluation

Supposing we could build the hybrid base-meta-level AI system for ethical behavior, how would we evaluate it? This can be addressed in multiple ways. First, from a CC point of view, we would want to know if the system is *creative*. How to establish this is still an open question, but there are several approaches to evaluation of CC systems that have been proposed. Collectively, these can examine both system product and process and include Ritchie's suggestions for formally stated empirical criteria focusing on the relative value and novelty of system output (2007); the FACE framework for qualifying different kinds of creative acts performed by a system (Colton, Charnley, and Pease 2011); the SPECS methodology which requires evaluating the system against standards that are drawn from a system specification-based characterization of creativity (Jordanous 2012); and Ventura's proposed spectrum of abstract prototype systems that can be used as landmarks by which specific CC systems can be evaluated for their relative creative ability (2016).

Second, from a behavioral point of view, we would want to know a) if the system's behaviors are *ethical* and b) if the system's behaviors are *useful*. Given that the main argument here concerns ethical behavior, the former must be the point of focus, but, given that, the latter will bear evaluation as well. Evaluating the ethics of such system behaviors is no more or less difficult than it is with extant AI systems or with humans.[9] Evaluating the utility of system behaviors is a well-understood problem and can be addressed using traditional AI evaluation methods, given a particular measure of utility.

Third, from an ethical point of view, we would want to *comprehend* the ethics of the system. Interestingly, given that the proposed system includes a meta-level for inventing normative ethics, this suggests the idea of developing a descriptive ethics for such AI systems. For obvious reasons, this is likely to be somewhat easier than doing so for human subjects, and at the same time, it is possible that the empirical study of populations of ethical AI systems could shed light on human ethics as well. For example, it is not difficult to imagine a large population of agents, all of whom possess the same meta-level ethics, admitting an empirically derived, potentially comprehensive description of that meta-level ethics. If that meta-level ethics is an operationalization of a cognitively plausible approach to ethics, one *might* be able to draw dependable conclusions about a human population operating under the meta-level ethics in question. Or, we might imagine scenarios involving multiple groups of agents, where each group possesses a different meta-level ethics, admitting the possibility of *differential* descriptive ethics that would likely be impossible with human subjects yet might yield conclusions that at least partially translate to such subjects.

Additional Considerations

There are many other interesting angles to consider here. For example, so far we have implicitly assumed that it is possible to create a domain-independent ethics. That is, given a meta-level ethics, an agent can use this as an aesthetic for creating a normative ethics that can then be applied as an aesthetic for judging candidate actions, *independent of the domain in which those actions may be applied.* The reality of *applied* ethics suggests that this assumption is likely incorrect—that rather than having a meta-level system that creates normative ethics, we should be thinking about a meta-level system that creates applied ethics. This means that the agent's environment (in a very general sense) must somehow inform either the aesthetic or the meta-aesthetic (or possibly both). Perhaps the meta-level can still produce a normative ethics and the base-level aesthetic can somehow specialize this appropriately for the domain of application. Or, perhaps the

[9] That is to say, this is likely even more difficult than addressing the question of the system's creativity.

meta-aesthetic must incorporate the domain of application, producing directly an applied ethics as its output artifact. It is, of course, possible that the same concern applies at the meta-level and that we can not even hope for a domain-independent meta-level ethics, but for now we will ignore this.

Another interesting consideration is the social aspect of ethics. Jennings makes a rather elegant argument about the social aspects of creativity and how, somewhat paradoxically, autonomous creativity *requires* significant social interaction (2010). Because his arguments center on the aesthetic judgment of the agent, they can be somewhat readily applied to our current discussion. He proposes that an agent in a social setting will not only have a model of its own aesthetic but also will have a model of its beliefs about other agents' aesthetics; it is in the dynamic updating of these models, due to social interactions, that the agent can develop true autonomous creativity; and, these social interactions are driven by psychologically plausible mechanisms such as propinquity, similarity, popularity, familiarity, mutual affinity, pride, cognitive dissonance, false inference and selective acceptance seeking. Because we are proposing ethics as aesthetic, we can follow a similar train of thought—an agent can model not only its own ethics but also (its perception of) those of all other agents. Social interaction can be a driving force behind the evolution of ethics, both at the individual and at the group level.

Yet another area for further study is the computational tenability of the proposed approaches. There is a rather simple argument for why the general problem of CC may not be computable that hinges on the decidability of the aesthetic (Ventura 2014). If the aesthetic *is* decidable, then the problem of generating candidate artifacts and filtering them with the aesthetic is computable (though efficiency could certainly still be an issue); however, if the aesthetic is *not* decidable, there is a simple reduction from the halting problem that shows that the creation of artifacts is not computable (in the theoretical computer science sense). This means that any operationalized ethics or meta-level ethics must be decidable, and given the nature of ethics, it is not clear how onerous a requirement this may be.[10]

Conclusion

We've proposed an appeal to computational creativity that addresses the problem of ethical agent behavior, which to our knowledge is a new way to look at the problem—suggesting a base-level system for which ethics is employed as an aesthetic for selecting behaviors coupled with a meta-level system for which meta-level ethics is employed as a meta-aesthetic for selecting ethics. This approach is, additionally, a new application of computational creativity, as, to date, no systems have been proposed for creating in the abstract domain of general behavior, nor, in particular, in the domain of ethics. While the current work is a position statement that asks many more questions than it answers, we believe the ethics-as-aesthetic approach to the problem of ethical agent behavior offers at least one, and possibly the only, way forward.

[10]Is it possible that recognizing an ethical action is "easy" while recognizing an unethical action is "hard"? Perhaps society itself accepts as ethical those actions that everyone deems ethical and rejects as unethical those that no one deems ethical but isn't sure about those with mixed reception. Any operationalized ethics that accurately models such a scenario will not be decidable given the existence of all three types of action.

References

Anderson, M., and Anderson, S. L., eds. 2006. *Special Issue on Machine Ethics*, volume 21(4). IEEE Intelligent Systems.

Anderson, M., and Leigh, S., eds. 2011. *Machine Ethics*. Cambridge University Press.

Anderson, M.; Anderson, S.; and Armen, C. 2005. Towards machine ethics: Implementing two action-based ethical theories. Technical report, *AAAI Fall Symposium*. 1-7.

Asimov, I. 1950. *I, Robot*. Bantam Books.

Audi, R. 2009. *Business Ethics and Ethical Business*. Oxford University Press.

Bickerman, G.; Bosley, S.; Swire, P.; and Keller, R. M. 2010. Learning to create jazz melodies using deep belief nets. In Ventura, D.; Pease, A.; Pérez y Pérez, R.; Ritchie, G.; and Veale, T., eds., *Proceedings of the International Conference on Computational Creativity*, 228–237.

Binsted, K., and Ritchie, G. 1994. A symbolic description of punning riddles and its computer implementation. In *Proceedings of the Association for the Advancement of Artificial Intelligence*, 633–638.

Bostrom, N., and Yudkowski, E. 2014. The ethics of artificial intelligence. In Frankish, K., and Ramsey, W. M., eds., *The Cambridge Handbook of Artificial Intelligence*. Cambridge University Press. 316–334.

Coeckelbergh, M. 2014. Sure, artificial intelligence may end our world, but that is not the main problem. *WIRED*.

Colton, S., and Wiggins, G. A. 2012. Computational creativity: The final frontier? In *Proceedings of the 20th European Conference on Artificial Intelligence*, 21–26. IOS Press.

Colton, S.; Bundy, A.; and Walsh, T. 1999. HR: Automatic concept formation in pure mathematics. In *Proceedings of the 16th International Joint Conference on Artificial Intelligence*, 786–791.

Colton, S.; Charnley, J.; and Pease, A. 2011. Computational creativity theory: The FACE and IDEA descriptive models. In *Proceedings of the 2nd International Conference on Computational Creativity*, 90–95.

Colton, S. 2012. The Painting Fool: Stories from building an automated painter. In McCormack, J., and D'Inverno, M., eds., *Computers and Creativity*. Berlin, Germany: Springer-Verlag. 3–38.

Cook, M.; Colton, S.; and Gow, J. 2016. The ANGELINA videogame design system, part I. *IEEE Transactions on Computational Intelligence and AI in Games* to appear.

European Parliament, Committee on Legal Affairs. 2017. *Draft Report with Recommendations to the Commission on Civil Law Rules on Robotics*. European Commission. Retrieved January 12, 2017.

Fox, S. 2009. Evolving robots learn to lie to each other. *Popular Science*.

Gates, D.; Agle, B. R.; and Williams, R. N. 2018. Teaching business ethics: Current practice and future directions. In Heath, E.; Kaldis, B.; and Marcoux, A., eds., *The Routledge Companion to Business Ethics*. Routledge. 60–76.

Jennings, K. E. 2010. Developing creativity: Artificial barriers in artificial intelligence. *Minds and Machines* 20(4):489–501.

Jones, T. 1991. Ethical decision making by individuals in organizations: An issue-contingent model. *The Academy of Management Review* 16(2):366–395.

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.

Kagan, S. 1998. *Normative Ethics*. Oxford: Westview Press.

Kant, I. 1994. *Ethical Philosophy*. Indianapolis: Hackett. Trans. James W. Ellington.

Kurzweil, R. 2005. *The Singularity is Near*. Penguin Books.

Liapis, A.; Yannakakis, G. N.; and Togelius, J. 2012. Adapting models of visual aesthetics for personalized content creation. *IEEE Transactions on Computational Intelligence and AI in Games* 4(3):213–228.

Lin, P.; Bekey, G.; and Abney, K. 2008. *Autonomous Military Robotics: Risk, Ethics, and Design*. US Department of Navy, Office of Naval Research.

McGee, G. 2007. A robot code of ethics. *The Scientist*.

Morris, R.; Burton, S.; Bodily, P.; and Ventura, D. 2012. Soup over bean of pure joy: Culinary ruminations of an artificial chef. In *Proceedings of the 3rd International Conference on Computational Creativity*, 119–125.

Muehlhauser, L., and Helm, L. 2012. Intelligence explosion and machine ethics. In Eden, A.; Søraker, J.; Moor, J. H.; and Steinhart, E., eds., *Singularity Hypotheses: A Scientific and Philosophical Assessment*. Berlin: Springer.

Müller, V. C. 2016. *Risks of Artificial Intelligence*. CRC Press - Chapman & Hall.

Norton, D.; Heath, D.; and Ventura, D. 2013. Finding creativity in an artificial artist. *Journal of Creative Behavior* 47(2):106–124.

Oliveira, H. G. 2012. PoeTryMe: a versatile platform for poetry generation. In *Proceedings of the ECAI 2012 Workshop on Computational Creativity, Concept Invention, and General Intelligence*.

Özbal, G.; Pighin, D.; and Strapparava, C. 2013. BRAINSUP: Brainstorming support for creative sentence generation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, 1446–1455.

Pachet, F., and Roy, P. 2014. Non-conformant harmonization: the real book in the style of Take 6. In *Proceedings of the 5th International Conference on Computational Creativity*, 100–107.

Pérez y Pérez, R., and Sharples, M. 2004. Three computer-based models of storytelling: BRUTUS, MINSTREL and MEXICA. *Knowledge-Based Systems* 17(1):15–29.

Riedl, M. O., and Young, R. M. 2010. Narrative planning: Balancing plot and character. *Journal of Artificial Intelligence Research* 39(1):217–268.

Ritchie, G. 2007. Some empirical criteria for attributing creativity to a computer program. *Minds and Machines* 17:67–99.

Sandel, M. 2009. *Justice: What's the Right Thing to Do?* Farrar, Straus and Giroux.

Smith, M. R.; Hintze, R. S.; and Ventura, D. 2014. Nehovah: A neologism creator nomen ipsum. In *Proceedings of the 5th International Conference on Computational Creativity*, 173–181.

Stock, O., and Strapparava, C. 2003. HAHAcronym: Humorous agents for humorous acronyms. *Humor - International Journal of Humor Research* 16(3):297–314.

Strapparava, C.; Valitutti, A.; and Stock, O. 2007. Automatizing two creative functions for advertising. In *Proceedings of 4th International Joint Workshop on Computational Creativity*, 99–105.

Toivanen, J. M.; Toivonen, H.; Valitutti, A.; and Gross, O. 2012. Corpus-based generation of content and form in poetry. In *Proceedings of the 3rd International Conference on Computational Creativity*, 175–179.

Varshney, L.; Pinel, F.; Varshney, K.; Schorgendorfer, A.; and Chee, Y.-M. 2013. Cognition as a part of computational creativity. In *Proceedings of the 12th IEEE International Conference on Cognitive Informatics and Cognitive Computing*, 36–43.

Veale, T., and Hao, Y. 2007. Comprehending and generating apt metaphors: A web-driven, case-based approach to figurative language. In *Proceedings of the 22nd AAAI Conference on Artificial Intelligence*, 1471–1476.

Veale, T. 2013. Less rhyme, more reason: Knowledge–based poetry generation with feeling, insight and wit. In Maher, M. L.; Veale, T.; Saunders, R.; and Bown, O., eds., *Proceedings of the Fourth International Conference on Computational Creativity*, 152–159.

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*, 208–217. LNAI 8598.

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*, 17–24.

Ventura, D. 2017. How to build a cc system. In *Proceedings of the 8th International Conference on Computational Creativity*, 253–260.

Wallach, W., and Allen, C. 2008. *Moral Machines: Teaching Robots Right from Wrong*. USA: Oxford University Press.

Creative Invention Benchmark

Matthew Guzdial, Nicholas Liao, Vishwa Shah, and Mark O. Riedl
School of Interactive Computing
Georgia Institute of Technology
Atlanta, GA 30332 USA
mguzdial3@gatech.edu, nliao7@gatech.edu, vishwashah@gatech.edu, riedl@cc.gatech.edu

Abstract

In this paper we present the Creative Invention Benchmark (CrIB), a 2000-problem benchmark for evaluating a particular facet of computational creativity. Specifically, we address combinational p-creativity, the creativity at play when someone combines existing knowledge to achieve a solution novel to that individual. We present generation strategies for the five problem categories of the benchmark and a set of initial baselines.

Introduction

Benchmarks represent a common means for driving community effort on a specific task. For example, MNIST is a dataset of handwritten digits paired with their numeric value, which proved popular and produced breakthroughs in the field of image recognition (LeCun 1998). At present it is considered solved by modern methods, but has continued as a standard for evaluating novel approaches, given that there are known performances for comparable techniques. While imperfect, we posit a benchmark for creativity could accomplish similar effects for the field of computational creativity. Creativity itself is too ill-defined for any benchmark. But just as recognizing handwritten digits does not translate to mastery of computer vision, we can define a set of creative tasks to address one particular facet of creativity.

Imagine you are an agent with a limited knowledge base. You know about the color red (255,0,0) and blue (0,0,255), and you know that there are integer variables x and y that can range from 0 to 100. You have access to a single method **Paint**, that given a value for x and y, and a color (represented in RGB) paints a pixel of a canvas. In return for painting a pixel the agent receives a floating point score [0-1] that grades the agents current painting compared to an unseen goal. The goal, in this case, is to paint a picture of a grape.

As a human reading the problem description above, the answer to this problem appears obvious. From red and blue make purple, or perhaps multiple shades of purple, in order to paint the unseen picture of a grape. This instinct can be understood as an instantiation of what Boden calls combinational creativity (Boden 2004). But it does not reflect how a naive AI agent might solve this problem, instead greedily placing blue and red to maximize the score to some local maximum without inventing the color purple. Alternatively one might naively hand-author all possible colors for the AI agent, but this would run against the spirit of the problem. Solving this problem clearly requires creativity, but we do not argue that this problem can evaluate the entirety of creativity. We instead focus on *p-creative*, *combinational* creativity (Boden 1998). P-creativity refers to creation of artifacts that are novel to the individual creator based on its knowledge (e.g., the artifact could have been invented by other creators previously). Combinational creativity refers to the creation of artifacts through the process of recombining existing knowledge. For the purposes of this paper we refer to this class of problem as invention problems.

In this paper we present the Creative Invention Benchmark (CrIB)[1], a publicly available benchmark of 2000 problems in 5 domains (painting, alien language, photobashing, narrative, and dessert recipes). All of these problems fit the general form of the painting example, requiring an agent to generalize and invent new concepts from a given problem-specific knowledge base to reach a solution given feedback from an unseen goal.

The example of painting an unseen grape may seem trivial but it is analogous to many of the most interesting and practical problems currently facing society from product invention to drug discovery. As humans we reflect on our existing knowledge to invent radical solutions, and we anticipate a need for artificial agents to do the same.

An important—but largely overlooked—challenge in computational creativity is cross-domain creativity, wherein a single agent or model is able to address creative problems from disparate domains. Commonly creativity researchers sidestep the need for general creative reasoning through hand-authoring of domain-specific knowledge. To the best of our knowledge this represents the first such cross-domain benchmark for computational creativity.

The rest of this paper is organized as follows: In section two we discuss related work and historic work that informs our position for this paper. In section three we discuss CrIB, all five problem categories and examples of each problem. In section four we demonstrate various baselines and features of the benchmark. We end with a discussion of the limitations of the benchmark, applications, and future directions.

[1] https://github.com/mguzdial3/CrIB

Related Work

Creativity Tests

There exist prior formal tests that involve computational models of creativity. For example the Lovelace 1.0 (Bringsjord, Bello, and Ferrucci 2003), Lovelace 2.0 (Riedl 2014) and MacGyver tests (Sarathy and Scheutz 2017) formalize bounds and loose evaluations that require creative cognition. However none of these prior approaches present sets of individual problems. Ravens Progressive Matrices (Raven and others 1938) has been used as a test for general cognitive ability, which includes creativity, most notably in work such as (Shegheva and Goel 2018). However, this test does not specifically seek to test creativity and only makes use of a single domain, whereas CrIB focuses on cross-domain, combinational p-creativity.

Combinational Creativity

There exists a range of combinational creativity techniques, which we briefly summarize. Notably researchers of combinational creativity do not frequently self-identify as addressing the same problem or field. Thus many combinational creativity approaches remain dependent on particular problem domains. However there has been some recent work to attempt to tie this field together (Guzdial and Riedl 2018).

Case-based reasoning (CBR) represents a general AI problem solving approach that relies on the storage, retrieval, and adaption of existing solutions (De Mantaras et al. 2005). The adaption function has lead to a large class of combinational creativity approaches, falling in two categories of either substitutional or structural adaption (Wilke and Bergmann 1998; Fox and Clarke 2009). These techniques tend to be domain-dependent, for example for the problem of text generation or tool creation (Hervás and Gervás 2006; Sizov, Öztürk, and Aamodt 2015).

Genetic Algorithms (GAs) represents a general AI problem solving approach that relies on an abstracted model of biological evolution (Srinivas and Patnaik 1994). It has proven extremely popular among computational creativity practioners, and we make use of it for an initial agent for solving CrIB. While not often recognized as such, the crossover function of a GA can be understood as a combinational creativity approach (Herrera, Lozano, and Sánchez 2003), though as with CBR adaption crossover functions tend to be domain-dependent.

Beyond CBR and GAs the area of belief revision, modeling how beliefs change, includes a function to merge existing beliefs with new beliefs(Konieczny, Lang, and Marquis 2004; Steels and De Beule 2006; Cojan and Lieber 2008; 2009; Konieczny and Pérez 2011). The mathematical notion of convolution has also been applied to blend weights, but with inconclusive results (Thagard and Stewart 2011).

We identify three combinational creativity approaches for further discussion given their popularity and generality across multiple problem domains. We visualize these approaches with illustrative examples in Figure 1.

Concept Blending Fauconnier and Turner (1998) formalized the "four space" theory of concept blending. They described four spaces: two *input spaces* represent the unblended elements, input space points are projected into a common *generic space* to identify equivalence, and these equivalent points are projected into a *blend space*. In the blend space, novel structure and patterns arise from the projection of equivalent points. Fauconnier and Turner (Fauconnier and Turner 1998; 2002) argued this was a ubiquitous process, occurring in discourse, problem solving, and general meaning making.

Concept blending typically requires a large amount of human authoring for individual concept spaces. More recent work has looked into automatically learning or deriving concepts (O'Donoghue et al. 2015; Guzdial and Riedl 2016). There has been work in blending individual tagged exemplars together based on surface level features of components (Alhashim et al. 2014). Fauconnier and Turner originally developed a set of heuristics for domain-independent measures of quality for blends, while more recent work has looked to introduce goals for blends (Li et al. 2012).

Amalgamation Ontañón and Plaza designed amalgams as a formal unification function between multiple cases (Ontañón and Plaza 2010). Similar to concept blending, amalgamation requires a knowledge base that specifies when two components of a case share a general form, for example "French" and "German" both share the more general form "nationality". Unlike concept blending, this shared generalization does not lead to a merging of components, but requires that only one of components be present in a final amalgam. For example, a "red French car" and an "old German car" could lead to an "old red French car" or an "old red German car".

Amalgams have been utilized as the adaption function in CBR systems (Manzano, Ontanón, and Plaza 2011), combined with concept blending for product development (Besold and Plaza 2015), and adapted to an asymmetrical form for story generation (Ontanón, Zhu, and Plaza 2012). Amalgamation represents a strong general method for combinational creativity. However it suffers from the drawbacks of other methods in terms of a traditional reliance on authored knowledge bases and domain-specific generalization.

Compositional Adaption Compositional adaption arose as a CBR adaption approach (Holland 1989; Fox and Clarke 2009), but has found significant applications in adaptive software (McKinley et al. 2004; Eisenbach, Sadler, and Wong 2007). The intuition behind compositional adaption is that individual concept components can be broken apart and recombined based on their connections. In adaptive software this process takes sets of functions with given inputs and outputs, and strings them together to achieve various effects, which makes compositional adaption similar to planning given a goal state or output. However, it can also be applied in a goal-less way to generate valid compositions.

Compositional adaption has been applied to recipe generation (Müller and Bergmann 2014; Badie and Mahmoudi 2017), intelligent tutoring systems (Reyhani, Badie, and Kharrat 2003), and traditional CBR approaches (Chedrawy and Abidi 2006). Unlike other methods compositional adaption does not require an explicit generalization knowledge

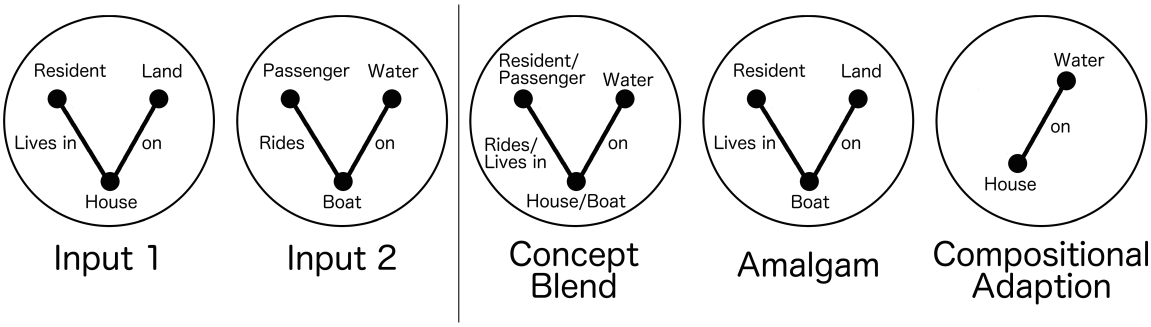


Figure 1: Example of three combinational creativity techniques. Two input spaces on left with example output from the three techniques on the right.

base. However, it is common to make use of a knowledge base to generalize across components and their relationships in order to expand the set of valid combinations.

Creative Invention Benchmark (CrIB)

In this section we discuss in more detail the Creative Invention Benchmark (CrIB). Our goal for the benchmark is to evaluate goal-driven combinational p-creativity, meaning a creative problem solving technique that relies on recombining the knowledge available to an individual. We refer to this class of problems as invention problems. To address our goal of generality we test combinational p-creativity across five distinct domains. This further reflects the multidisciplinary field of computational creativity. The domains are:

1. **Painting**, as in the running example in the introduction, in which an agent must invent new colors from some initial knowledge base to approximate some unknown goal painting.
2. **Alien language**, in which an agent must invent novel words to recreate an unknown goal sentence.
3. **Photobashing**, a practice from the field of concept art in which existing images are pieced together to form novel art. In this problem domain the agent must combine input images to approximate some unknown goal photobash.
4. **Narrative**, in which an agent, given a graphical representation of at least two story domains, must tell a target unknown goal story in some novel domain.
5. **Dessert Recipes**, in which an agent must combine existing recipe ingredients to create an unknown goal recipe.

The benchmark has a total of 2000 problems evenly spread across the five domains for a total of 400 problems per domain. For each problem an agent receives an initial knowledge base, a function to apply the agent's knowledge base in a domain-appropriate way (e.g. adding words to a sentence in the alien language domain), a function to clear the current state of the agent's submission in a domain-appropriate way (e.g. resetting the current canvas to a blank canvas in the painting domain), and a scoring function that measures the agent's distance to some unknown goal (with values ranging from 0.0 to 1.0).

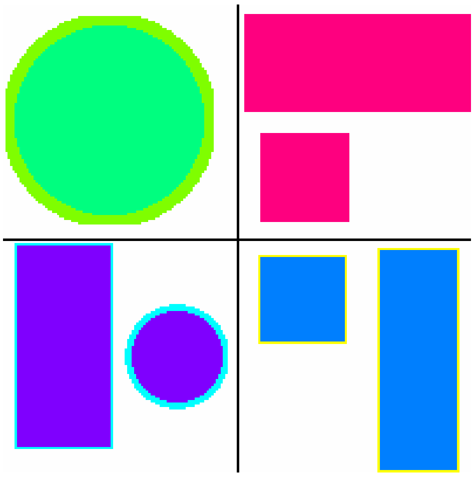


Figure 2: Four examples of unseen goal "paintings"

In the following subsections we discuss each domain in further detail. Notably we discuss the structure of each problem in terms of the input knowledge base, functions available to the agent, and the problem goal. We also discuss the approach taken to generate the domain problems and demonstrate an example problem. We note that all relevant code can be found at the public CrIB GitHub.

Painting

We include painting as a domain due to the long history of visual art in computational creativity, such as AARON (Cohen 1995) and the Painting Fool (Colton 2012). The painting problems of CrIB reflect the general problem description outlined in the introduction.

- **Input:** 2-6 colors as an initial knowledge base or palette.
- **Goal:** A painting that includes colors not included in the agent's initial knowledge base. The agent cannot directly access this goal painting.
- **Domain-Specific Function:** The agent is given a function **Paint** that takes as arguments two variables x and y (ranging from 0 to 1) that determine the location of a pixel and

Figure 3: Example of a human photobash from one of our artists on the right with the input images used on the left.

Figure 4: Example randomly selected generated photobash on the right, with the input images on the left.

a color (represented in RGB format). This function then sets that pixel to the specified color on an initially blank canvas of fixed size.

- **Clear:** This function allows the agent to clear the current canvas, resetting it to an all-white image.
- **Score:** The scoring function compares the current canvas with the target painting running vector subtraction for each pixel, it then sums over these values and normalizes to a maximum of 1.0. Without an agent inventing new colors, it is impossible to score a perfect 1.0 on any of these problems. However, it is possible to get to a relatively large local maxima.

We present examples of target images in Figure 2. To generate these problems we wrote a simple combination process that takes the primary (red, green, blue), secondary, and tertiary colors of a color wheel and finds all possible additive and subtractive combinations of colors (e.g. red and blue to make purple) From there it selects either a single color combination or multiple color combinations to draw upon for each problem, creating random geometric shapes with the target colors. We sorted the final 400 questions in terms of the number of initial colors and the number of colors and shapes in the target image as a stand-in for difficulty.

Alien Language

We include a fictional alien language as one of our domains as a stand-in for many language domains in the field of computational creativity such as human and musical language. In addition, making use of an alien language allowed us to include one problem domain in which the answers would be less obvious to a human, given that the language would follow artificial rules without a basis in real language. This allows us to consider a human subject study as a future baseline.

- **Input:** A set of 3-9 words as an initial knowledge base or vocabulary.
- **Goal:** A sentence that includes words not in the initial vocabulary, representenced as a sequence of words. This sentence is not directly accessible to the agent.
- **Domain-Specific Function:** The agent is given a function **AddWord** that takes as arguments one word from the knowledge base and adds it to the current sentence.
- **Clear:** This function allows the agent to clear the current sentence, resetting it to an empty sequence.
- **Score:** The scoring function compares the current sentence with the target sentence, giving a score of 1.0 for a perfect match, a 0.0 for no match, and a proportional score for partial matches of words in order in the sentences.

The alien language problems were generated by first generating a 2000-word vocabulary composed of randomly composing words from the characters ‘A’, ‘B’, ‘C’, ‘D’, ‘W’, ‘X’, ‘Y’, and ‘Z’ varying in length between two and twelve characters. From there we made use of arbitrary rules to compose a total of 400 target sentences varying in length between two and five words. For each target sentence we found one or two words in the sentence that could be considered combinations of between two and three other words in the vocabulary. For example “WAZZ” could be broken into ‘WA’ and ‘ZZ’. We then sorted each problem according to the number of input words and its length as a stand-in for difficulty. For example a simple sentence might be “WAZZ BYXBYW XDWB” with the initial knowledge base ‘BYXBYW’, ‘XDWB’, ‘WA’, and ‘ZZ’.

Photobashing

Photobashing is the practice of combining sets of input images to create a new composite image. It is a common practice for concept and key art for films, television shows, and video games. We include photobashing as it fits our general problem format and represents a real-world application of combinational creativity.

- **Input:** A set of 2-9 images as an initial knowledge base.
- **Goal:** A goal image that represents a combination of the input images. This image or photobash is not directly accessible to the agent.

- **Domain-Specific Function:** The agent is given a function **Stamp**, which takes as arguments x and y variables (ranging from 0.0 to 1.0) and one of the images of the knowledge base and places this image at an x,y location of an initially blank canvas of fixed size. Note that the agent can only add entire images from its knowledge base, meaning invention must occur to reach the goal.
- **Clear:** This function allows the agent to clear the current canvas, resetting it to a blank canvas.
- **Score:** The same as the painting scoring function.

To start generating photobashes we first gathered a palette of over eighty royalty-free stock images and photographs. We then made use of two distinct approaches to combine these images. For one we asked five human artists of a range of skill to construct photobashes. This lead to a total of 80 photobashes with a median value of 14 photobashes contributed across the five artists. An example of a human photbash can be found in Figure 3. For the remaining 320 photobashes we constructed a simple visual grammar by breaking apart a number of images of animals into heads, torsos, front legs and back legs. We then ran a script to combine these components. We required a human to verify the coherency of each generated photobash to ensure a baseline of quality. We reran the generation process for each rejected photobash. An example of a generated photobash can be found in Figure 4. The problems were sorted according to number of input images used to construct the goal as a stand-in for difficulty.

Narrative

We include narrative as a problem domain as it represents a common area of creativity research and allows us to include a novel representation. We made use of a story or plot graph representation as it encodes the branching nature of stories (Weyhrauch 1997). Plot graphs can be understood as a directed graph with the nodes as story events and the edges representing preconditions for events. Plot graphs represent multiple possible stories in a given domain and can generate stories by walking the graph.

- **Input:** A set of 2-4 distinct plot graphs
- **Goal:** A goal story represented as a sequence of events that cannot be generated from any of the input plot graphs. This story is not directly accessible to the agent.
- **Domain-Specific Function:** The agent is given a function **Submit**, which takes a single plot graph argument, and finds the closest story in the graph to the goal story. This closest story is set as the current story.
- **Clear:** This function removes any current story.
- **Score:** This function compares the current story and the target story. It returns 1.0 if the two match exactly, 0.0 if the two completely differ, and otherwise a proportional score for the number of shared events in sequence.

To begin the generation of narrative problems we first encoded ten existing published plot graphs in a common representation. We did this to ensure we did not accidentally encode too much stylistic similarity in the plot graphs. We pulled the movie, robbery, and pharmacy plot graphs from (Li 2015), the cat lover, cattle driver, stage coach and tour bus plot graphs from (Permar and Magerko 2013), the inheritance plot graph from (Min et al. 2008), the fantasy plot graph from (McIntyre and Lapata 2010), and the horror Anchorhead plot graph from (Nelson and Mateas 2005). For each plot graph we replaced the names of characters, each only had up to two, with 'A' and 'B'. We also simplified a few of the plot graphs such that each were at most 20 nodes. We then made use of amalgamation (Ontañón and Plaza 2010) to generate new plot graphs. To allow for mapping across different plot graphs we hand tagged certain event nodes with a higher-order theme (e.g. 'intro', 'ending', etc), additionally allowing mapping on shared words across nodes. From these plot graph amalgams we generated stories, which we then hand-checked to ensure coherency. For example a combination of fantasy and tourbus might output: "Monster holds B captive. A slays monster. A rescues B. A departs with B. A and B get married. A and B visit a Landmark." We sorted these problems according to the number of initial plot graphs used to create the goal story's plot graph.

Dessert Recipe

For our final domain we chose recipes, more specifically dessert recipes, as recipes represent a common example domain for adaption and creativity. This also allowed for a second real-world domain beyond photobashing. For each dessert recipe problem the agent must invent a recipe given existing recipes.

- **Input:** A set of 3-130 distinct recipes encoded as a recipe name and a set of ingredients (e.g. banana muffins (bananas, flour, eggs, milk, sugar)).
- **Goal:** A goal recipe distinct from all of the input recipes. This goal recipe is not directly accessible to the agent.
- **Domain-Specific Function:** The agent is given a function **Submit**, which takes a single recipe argument. This is set as the current recipe.
- **Clear:** This function removes any current recipe.
- **Score:** This function compares the current and target recipe ingredients. It returns a value between 0 and 1 dependent on the extent to which the two sets overlap.

To generate these problems we drew on the dessert dataset from (Veale 2017). For each dessert we found all sets of other desserts whose ingredients could be composed to match its ingredients. From this point it was simple to randomly select a set of four hundred of these possible compositions for each problem. We then sorted these problems according to the number of initial desserts in the knowledge base as a stand-in for difficulty. This number varied massively from 3 to 142. As an example given banana muffins (bananas, flour, eggs, milk, sugar), Vanilla wafer cake (shredded coconut, flour, milk, eggs, sugar, chopped pecans, vanilla essence), and treacle tart (golden syrup, lemon zest, butter, flour) produce pound cake (butter, sugar, eggs, flour, vanilla essence).

Table 1: Average output of two baselines and the random agent for each domain and across all five domains.

	Painting	Language	Photobash	Narrative	Dessert	Total
Null	0.70	0.0	0.76	0.0	0.0	0.29
Uncreative Max	0.85	0.72	0.89	0.45	0.49	0.61

Table 2: Scores for the presented agents and their average total.

	Painting	Language	Photobash	Narrative	Dessert	Total
Random Agent	-0.99	-2.42	-1.50	-0.32	-0.50	-1.15
GA_{100}	-0.99	-1.41	0.02	0.76	0.35	-0.25
GA_{1000}	-0.91	-1.19	0.17	0.81	0.35	-0.14

Using CrIB

In this section we discuss how to make use of CriB. We introduce two baselines to better characterize the benchmark, introduce a scoring function that relies on one of these two baselines, and present two initial agents that attempt to solve the benchmark.

Baselines

In this section we demonstrate two baselines to further characterize CrIB. The baseline "null" represents the score of an agent that does absolutely nothing. The baseline "Uncreative Max" represents the best an agent could do without any invention of additional knowledge beyond the initial input knowledge base for each problem. We constructed Uncreative Max by finding the closest element of the initial knowledge base to the target concepts.

We summarize the average scores of these two baselines in Table 1. We note that the two visual domains—painting and photobashing—can achieve the highest values since they only look at pixel-by-pixel comparisons and share many white pixels. In addition, it is relatively easy to score high on the alien language domain since the goal sentences are composed mostly of words from the initial knowledge base. However, narrative and dessert generation are far less successful. Our baselines are not meant to signify any intelligence, but to provide a means for analyzing how easy it is to guess a high-scoring solution without creative reasoning if we attempt to score naively.

Scoring Function

The prior section demonstrates that it is possible to get high scores without creative behavior if we score naively. However, we intend this benchmark to measure a facet of creativity. Therefore we use the following scoring function for each problem domain:

$$Score = (NScore_a - NScore_u)/(400 - NScore_u)$$

Where $Score$ represents our final score, $NScore_a$ represents the naive score discussed for each domain above for some current agent a, $NScore_u$ represents the naive score discussed above for the Uncertain Max baseline. In other words an agent's actual score is the amount that it does better than Uncreative Max. We are essentially making the assumption that if the score of Uncreative Max represents uncreative computation, whatever is left must require creative computation. Because Uncreative Max makes use of all available knowledge without any invention of new knowledge, an agent may receive a negative score if it fails to make use of all of the knowledge it is initially given.

Initial Agents

We present two initial agents as a means of demonstrating that the problems of this benchmark are non-trivial. For the first agent we present a random agent that randomly selects a single element of the initial knowledge base and runs the domain-specific function. We note that this first agent cannot be expected to do better than Uncreative Max, but we include it in order to compare it to our second agent. Our second agent is a genetic algorithm (GA) agent, which we tested in two variations.

The GA agent searches in the space of possibile final answers relevant to each domain (images for painting and photobashing, sentences for alien language, recipes for dessert recipe, and stories for narrative). It uses a mutation function that randomly swaps out some value of the current representation with a value from the knowledge base. It uses a crossover function that randomly selects values from two parents, selected according to current naive score, to fill in the variables of a new child representation (e.g. randomly grabbing words from two parent sentences to create a child sentence). We used a mutation rate of 0.7, and selected the 20 best parents to create 20 new children with each iteration. We created two variations on this agent based upon number of iterations and population size. For the first GA_{100} we ran the GA for a maximum of 100 iterations with a population of 100 individuals. For the second GA_{1000} we ran for a maximum of 1000 iterations with a population of 1000 individuals. We present the scores of all agents in Table 2.

We note a number of interesting results comparing the scores across these agents. GA_{1000} did the best, as one might expect, but did far worse than one might naively assume. The primary reason for this was that the simple mechanism by which both GA agents introduced new knowledge (random mutations and crossover) was insufficient to produce the desired combinations given the feedback of the scoring function. This is most clear in comparing GA_{1000} and GA_{100} in terms of the Dessert Recipes and Narrative performance. In the former there was no improvement in the score despite a tenfold increase in iterations and population.

The most successful domain was Narrative, since the agent's crossover and mutation functions were well-suited to swapping out events in a story. We found with additional tests that the GA_{1000} values largely represent the upper-bound of this approach, indicating that solving this benchmark is not simply a problem of longer training time.

Ways of Using CrIB

We include all of the discussed agents and baselines and a few additional agents on the public GitHub. Beyond reporting scores we recommend researchers make use of these given agents to draw comparisons. In particular beyond score we recommend reporting the average increase in size of the knowledge base per problem and the number of guesses or training steps necessary to achieve the reported scores. These features can allow for better comparison in terms of an agent's ability to make insightful or human-like combinations quickly. In terms of formats for reporting results we anticipate that this will depend on the agent. One clear approach would be to make use of Reinforcement Learning, which might involve reporting average score over time. Alternatively one might approach this problem with a more traditional classifier, at which point reporting training and testing error may be appropriate.

We note that one naive approach might be to hand-author knowledge for each domain. For example, simply giving an agent all primary, secondary, and tertiary colors for the painting domain. However, this goes against the spirit of the benchmark, and entirely removes any need for creative reflection or invention from an agent.

Limitations and Future Work

We note that the benchmark at present has a number of limitations. We do not present any successful, creative agents by our own measures in this paper. The development of such agents remains the largest area of future work Further, while relatively large at first glance 2000 problems is small compared to similar benchmarks in other domains. Notably it would be trivial to expand the painting, alien language, and the dessert recipe domains to many times their current size, which one can accomplish given the GitHub generator code. However the need for human evaluation for narrative and photobashing represents a limiting factor.

There are many more possible domains we could include in this benchmark. For example music and product generation, both common computational creativity domains. We fully intend to expand CrIB in future versions.

Conclusions

We present the Creative Invention Benchmark (CrIB), a benchmark for evaluating combinational p-creativity. We demonstrate the generative process for creating the 400 problems for each of the five domains of the benchmark, and the performance of a set of baselines and agents. We make this baseline available to the general research community through GitHub, and hope that it inspires further developments in the field of computational creativity.

Acknowledgments

We gratefully acknowledge the NSF for supporting this research under NSF award 1525967. We appreciate the detailed work of the ICCC reviewers, whose insight greatly improved this final paper. In addition we would like to especially thank Jack Yardley Ingram, without whom this paper would not have been finished.

References

Alhashim, I.; Li, H.; Xu, K.; Cao, J.; Ma, R.; and Zhang, H. 2014. Topology-varying 3d shape creation via structural blending. *ACM Transactions on Graphics (TOG)* 33(4):158.

Badie, K., and Mahmoudi, M. T. 2017. Compositional adaptation in case-based reasoning based on the semantic relations between the components in the cases. In *2017 IEEE International Conference on INnovations in Intelligent SysTems and Applications (INISTA)*, 449–454. IEEE.

Besold, T. R., and Plaza, E. 2015. Generalize and blend: Concept blending based on generalization, analogy, and amalgams. In *ICCC*, 150–157.

Boden, M. A. 1998. Creativity and artificial intelligence. *Artificial Intelligence* 103(1-2):347–356.

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

Bringsjord, S.; Bello, P.; and Ferrucci, D. 2003. Creativity, the turing test, and the (better) lovelace test. In *The Turing Test*. Springer. 215–239.

Chedrawy, Z., and Abidi, S. R. 2006. Case based reasoning for information personalization: using a context-sensitive compositional case adaptation approach. In *Engineering of Intelligent Systems, 2006 IEEE International Conference on*, 1–6. IEEE.

Cohen, H. 1995. The further exploits of aaron, painter. *Stanford Humanities Review* 4(2):141–158.

Cojan, J., and Lieber, J. 2008. Conservative adaptation in metric spaces. In *European Conference on Case-Based Reasoning*, 135–149. Springer.

Cojan, J., and Lieber, J. 2009. Belief merging-based case combination. In *ICCBR*, 105–119. Springer.

Colton, S. 2012. The painting fool: Stories from building an automated painter. In *Computers and creativity*. Springer. 3–38.

De Mantaras, R. L.; McSherry, D.; Bridge, D.; Leake, D.; Smyth, B.; Craw, S.; Faltings, B.; Maher, M. L.; T COX, M.; Forbus, K.; et al. 2005. Retrieval, reuse, revision and retention in case-based reasoning. *The Knowledge Engineering Review* 20(3):215–240.

Eisenbach, S.; Sadler, C.; and Wong, D. 2007. Component adaptation in contemporary execution environments. In *Distributed Applications and Interoperable Systems*, 90–103. Springer.

Fauconnier, G., and Turner, M. 1998. Conceptual integration networks. *Cognitive science* 22(2):133–187.

Fauconnier, G., and Turner, M. 2002. *The way we think: Conceptual blending and the mind's hidden complexities*. Basic Books.

Fox, J., and Clarke, S. 2009. Exploring approaches to dynamic adaptation. In *Proceedings of the 3rd International DiscCoTec Workshop on Middleware-Application Interaction*, 19–24. ACM.

Guzdial, M., and Riedl, M. 2016. Learning to blend computer game levels. In *Proceedings of the Seventh International Conference on Computational Creativity (ICCC 2016)*.

Guzdial, M., and Riedl, M. O. 2018. Combinatorial meta search. *Proceedings of the 1st Knowledge Extraction from Games Workshop*.

Herrera, F.; Lozano, M.; and Sánchez, A. M. 2003. A taxonomy for the crossover operator for real-coded genetic algorithms: An experimental study. *International Journal of Intelligent Systems* 18(3):309–338.

Hervás, R., and Gervás, P. 2006. Case-based reasoning for knowledge-intensive template selection during text generation. In *European Conference on Case-Based Reasoning*, 151–165. Springer.

Holland, J. H. 1989. *Induction: Processes of inference, learning, and discovery*. Mit Press.

Konieczny, S., and Pérez, R. P. 2011. Logic based merging. *Journal of Philosophical Logic* 40(2):239–270.

Konieczny, S.; Lang, J.; and Marquis, P. 2004. Da2 merging operators. *Artificial Intelligence* 157(1-2):49–79.

LeCun, Y. 1998. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*.

Li, B.; Zook, A.; Davis, N.; and Riedl, M. O. 2012. Goal-driven conceptual blending: A computational approach for creativity. In *Proceedings of the 2012 International Conference on Computational Creativity, Dublin, Ireland*, 3–16.

Li, B. 2015. *Learning knowledge to support domain-independent narrative intelligence*. Ph.D. Dissertation, Georgia Institute of Technology.

Manzano, S.; Ontanón, S.; and Plaza, E. 2011. Amalgam-based reuse for multiagent case-based reasoning. In *International Conference on Case-Based Reasoning*, 122–136. Springer.

McIntyre, N., and Lapata, M. 2010. Plot induction and evolutionary search for story generation. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, 1562–1572. Association for Computational Linguistics.

McKinley, P. K.; Sadjadi, S. M.; Kasten, E. P.; and Cheng, B. H. 2004. A taxonomy of compositional adaptation. *Rapport Technique numéroMSU-CSE-04-17*.

Min, W.-H.; Shim, E.-S.; Kim, Y.-J.; and Cheong, Y.-G. 2008. Planning-integrated story graph for interactive narratives. In *Proceedings of the 2nd ACM international workshop on Story representation, mechanism and context*, 27–32. ACM.

Müller, G., and Bergmann, R. 2014. Compositional adaptation of cooking recipes using workflow streams. In *Computer cooking contest, workshop proceedings ICCBR*.

Nelson, M. J., and Mateas, M. 2005. Search-based drama management in the interactive fiction anchorhead. In *AIIDE*, 99–104.

O'Donoghue, D. P.; Abgaz, Y.; Hurley, D.; and Ronzano, F. 2015. Stimulating and simulating creativity with dr inventor. In *Proceedings of the 6th ICCC*.

Ontañón, S., and Plaza, E. 2010. Amalgams: A formal approach for combining multiple case solutions. In *Case-Based Reasoning. Research and Development*. Springer. 257–271.

Ontanón, S.; Zhu, J.; and Plaza, E. 2012. Case-based story generation through story amalgamation. In *Proceedings of the ICCBR 2012 Workshops*, 223–232.

Permar, J., and Magerko, B. 2013. A conceptual blending approach to the generation of cognitive scripts for interactive narrative. In *Proceedings of the 9th AIIDE Conference*.

Raven, J. C., et al. 1938. *Raven's progressive matrices*. Western Psychological Services.

Reyhani, N.; Badie, K.; and Kharrat, M. 2003. A new approach to compositional adaptation based on optimizing the global distance function and its application in an intelligent tutoring system. In *Information Reuse and Integration, 2003. IRI 2003. IEEE International Conference on*, 285–290. IEEE.

Riedl, M. O. 2014. The lovelace 2.0 test of artificial creativity and intelligence. *arXiv preprint arXiv:1410.6142*.

Sarathy, V., and Scheutz, M. 2017. The macgyver test-a framework for evaluating machine resourcefulness and creative problem solving. *arXiv preprint arXiv:1704.08350*.

Shegheva, S., and Goel, A. 2018. The structural affinity method for solving the raven?s progressive matrices test for intelligence. In *Proceedings of Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-18)*.

Sizov, G.; Öztürk, P.; and Aamodt, A. 2015. Evidence-driven retrieval in textual cbr: bridging the gap between retrieval and reuse. In *International Conference on Case-Based Reasoning*, 351–365. Springer.

Srinivas, M., and Patnaik, L. M. 1994. Genetic algorithms: A survey. *computer* 27(6):17–26.

Steels, L., and De Beule, J. 2006. Unify and merge in fluid construction grammar. *EELC* 4211:197–223.

Thagard, P., and Stewart, T. C. 2011. The aha! experience: Creativity through emergent binding in neural networks. *Cognitive science* 35(1):1–33.

Veale, T. 2017. Déjà vu all over again.

Weyhrauch, P. 1997. *Guiding Interactive Fiction*. Ph.D. Dissertation, Ph. D. Dissertation, Carnegie Mellon University.

Wilke, W., and Bergmann, R. 1998. Techniques and knowledge used for adaptation during case-based problem solving. In *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*, 497–506. Springer.

Exploring the Engagement and Reflection Model with the Creative Systems Framework

Juan Alvarado[1] and **Geraint A. Wiggins**[1,2]
[1] School of Electronic Engineering and Computer Science, Queen Mary University of London, United Kingdom
[2] Department of Computer Science, Vrije Universiteit Brussel, Belgium
{j.alvaradolopez, geraint.wiggins}@qmul.ac.uk

Abstract

We employ the Creative Systems Framework (Wiggins 2006) to explore the Account of Writing as Creative Design proposed by Sharples (Sharples 1996). The purpose of this exploration is to have a deeper understanding of this proposal and so, be able to analyse computer implementations of it.

Conceptual spaces

Boden (1990) points out that there is a *Conceptual Space (CS)* where creative ideas exist. She suggests that this CS has origin in the culture of the creator and is any disciplined way of thinking that is familiar to (and valued by) a certain social group. Boden (1990) defines a CS as a structured style of thought and she points out that conceptual spaces are normally learned from the culture. For any CS there are rules or constraints which form it and in this CS new ideas (concepts) may be found.

Boden (1990) explains that concepts can be found in a CS by *Exploration* and *Transformation*. She states that by exploring the CS someone may be able to see possible concepts that had not been discovered yet. By transforming the CS its form changes because the rules or constraints have been changed and different concepts may be available to be found.

Writing as a creative design

Sharples (1996) proposes an account of writing as a creative design. Sharples points out that the main part of this account is that writing is a cognitive activity and an open-ended design process that requires tools, resources and setting constraints on goals, plans, etc., but creative writing also requires breaking of such constraints (Sharples 1996).

The writer imposes appropriate constraints that come from a combination of the task itself, external resources, knowledge and experience of the writer.

Sharples explains that it is also necessary to distinguish between novelty and adequacy and creativity to advance in the description of the mechanisms of creative writing. Boden (1990) in her analysis of cognition and creativity explains that in conceptual spaces it is possible to find new ideas. Sharples points out that those ideas, found in conceptual spaces, must be not only novel but also appropriate for the task and the public. Sharples explains that an important part of his account is the description of a set of constraints to generate appropriate content. They constraint the generative system and form what Boden (1990) describes as the conceptual space. A conceptual space limits the scope of the search through long-term memory to the concepts and schemes that are appropriate for the task. It can be restrictive and invoke a flow of conventional ideas, but also provides the source material for creativity (Sharples 1996).

A writing task begins with a given set of constraints. These can be external, such as a topic, previously written material, or a set of editor guidelines. They can also come from the writer, such as the schemes, interrelated concepts, genres and knowledge of a language that form the conceptual spaces of the writer. The task is also restricted by the tools a writer uses and by the context in which the writing occurs. These constraints act together to frame the activity of writing. The success of this task is how knowledge is guided by the restriction so that a successful writer invoke just the right schemes (Sharples 1996).

Sharples explains that there are similarities between the studies of cognition in design and the cognitive theories of creativity in writing. Sharples (1996) explains:

- Design problems are open-ended and can not be fully specified. They do not have a fixed set of goals or a sequence of steps, each of which can be evaluated in terms of their proximity to the goal.
- The design process is endless. There is an inexhaustible amount of possible solutions, and the end of the design process is a matter of criteria. A designer stops when it no longer seems worth the effort to try to improve the quality of the product, or by some external factor, such as running out of time or resources.
- There is no design process that is infallibly correct. There are many different and equally successful approaches, and good designers can control and vary their strategies according to the task.
- The process involves finding and solving problems. The design process does not consist of a clear sequence of stages prior to a finished product, and much of a designer's time is spent identifying and refining the problem. Sharples also summarises this idea by saying the problem is generated while it is being solved.

- Design inevitably implies a subjective value judgement. A designer asks questions and produces products that can only be judged by a subjective evaluation of quality.
- Design is a prescriptive activity. Unlike the process of scientific discovery, where the objective is to describe the world, design cares about what might, could and should be. It prescribes and creates the future, which requires ethical and moral scrutiny.

Primary generators, which are also constraints, are other important components in this account of writing. A primary generator is a powerful idea, but easy to fix that a designer uses to drive and guide the activity (Sharples 1996).

In a writing task, writers often have to manipulate knowledge externally. To achieve this they can use a number of tools, for example, paper or a computer to capture their mental representations in order to be able to modify, transform and order them. This depends on the resources that the writer has available and they are also part of the constraints the process has (Sharples 1996).

Sharples explains that it is important to observe the effect of the environment and tools on the writing task as a design.

Sharples explains that an episode of writing does not begin with a single goal, but with a set of external and internal constraints. These come as a combination of the task, a collection of resources, aspects of the knowledge and experience of the writer, and a primary generator.

As writing progresses, the constraints provide tacit knowledge to guide the writing process. The writer can re-represent some of them in a more explicit way, as a conceptual space to be explored and transformed. The movement between engaged writing, guided by tacit restriction, and more deliberate reflection forms the cognitive motor of writing (Sharples 1996).

Engagement An engaged writer devotes full attention to creating a chain of associated ideas and converting them into text. The working memory is completely dedicated to the task, and the only other deliberative mental activity that the writer can carry out in the text creation exercise is to speak the words out loud (Sharples 1996).

In order to reflect on the text, it is necessary to stop writing, and the result is that the periods of engagement are interleaved with periods of reflection (Sharples 1996).

Reflection Reflection consists of "sitting back" and reviewing all or part of the written material, conjuring up memories, generating ideas by association, forming and transforming ideas, and planning what new material to create and how to organise it. (Sharples 1996).

The cycle of engagement and reflection establishes distinctive rhythms that characterise writing processes. The period of these rhythms can be short, as when a writer looks back over each sentence, as it is written, or longer when a writer rereads an entire piece of writing and plans a thorough revision (Sharples 1996).

Sharples (1996) makes a special distinction between regular writing activity and explicit knowledge manipulation. He explains that it is possible, for example, to produce grammatically correct language without reciting the rules of grammar. But to explore and transform conceptual spaces is necessary to invoke constraints and schemas as explicit entities and work on them deliberately. Sharples (1996) explains that the mind exploits the knowledge that has already been stored, re-enacting tacit procedures as well as explicit structures. The representational redescription provides us with the means to reflect on the experience. It allows us to review an activity, re-cast it as a mental schema and use it to probe long-term memory, recall related schemas, integrate the new knowledge with the previous one and explore and transform it. Sharples (1996) explains that this transition from tacit knowledge to representational redescription is not easy, even for experienced writers.

In the next section the Creative Systems Framework (CSF) (Wiggins 2006) is explained and later this account of writing proposed by Sharples (1996) is analysed in terms of the CSF. Implementation examples of this account of writing explained are analysed with the resulting CSF framework.

Creative Systems Framework

Wiggins (2006) formalises the ideas on creativity expressed by Boden (1990). He argues that at first sight Boden's proposal lacks elements to use it in a consistent way, so he formalised the concepts in Boden's theory so they can be better applied.

Wiggins explains that artefacts are produced by a system (a creator), in a certain context, like P-creative acts explained by (Boden 1990) which are related to the creator's mind and a culture that is familiar to a certain social group. Wiggins points out that novelty and value are important features of artefacts produced by a system in its context and many authors coincide with this (e.g. (Boden 1990; Pérez y Pérez 1999; Ritchie 2007; Colton 2008)).

Wiggins defines different conceptual elements which are important in the analysis of a creative system.

Universe ($\mathcal{U}$) is a multidimensional space, whose dimensions are capable of representing anything and all possible distinct concepts correspond to distinct points in $\mathcal{U}$ (Wiggins 2006). Conceptual Spaces $\mathcal{C}$ defined by cultural agreements and for specific domains, in which concepts may exist, can be located inside the Universe $\mathcal{U}$.

Language ($\mathcal{L}$) is a common language from which framework's rules will be obtained.

Rules ($\mathcal{R}$) is a subset of $\mathcal{L}$ and are the rules which constrain a Conceptual Space $\mathcal{C}$; they define the nature of the created artefacts. In particular, in the societal context, they represent the agreed nature of what a concept is (Wiggins 2006).

Traversing strategy ($\mathcal{T}$) is a subset of $\mathcal{L}$ and is the set of rules which allow us to traverse the Conceptual Space ($\mathcal{C}$). $\mathcal{T}$ defines the way a particular agent produces an artefact in practical terms (Wiggins 2006).

Evaluation ($\mathcal{E}$) is a subset of $\mathcal{L}$ and is the set of rules for evaluation of concepts according to whatever criteria we may consider appropriate, they define the value of artefacts (Wiggins 2006).

The Creative Systems Framework proposal (Wiggins

2006) has some axiomatic points which are independent of the domain or type of the system.

Axiom 1 All possible concepts, including the empty concept, are represented in $\mathcal{U}$, so, $\top \in \mathcal{U}$.

Axiom 2 All concepts c_i represented in $\mathcal{U}$ are different, so, $\forall c_1, c_2 \in \mathcal{U}, c_1 \neq c_2$

Axiom 3 All conceptual spaces are strict subsets of $\mathcal{U}$, so, $\forall i \quad \mathcal{C}_i \subseteq \mathcal{U}$

Axiom 4 All conceptual spaces $\mathcal{C}$ include the empty concept $\top$, so, $\forall i \quad \top \in \mathcal{C}_i$

$\mathcal{R}$ represents the rules which define the nature of the created artefacts. So, $\mathcal{R}$ constraints the Conceptual Space ($\mathcal{C}$) suggested by Boden (1990). Wiggins (2006) explains that by using an interpretation function $[\![.]\!]$ it is possible to choose members of $\mathcal{U}$ which belongs to $\mathcal{C}$, assuming a well formed set $\mathcal{R}$. We get the expression: $\mathcal{C} = [\![\mathcal{R}]\!](\mathcal{U})$

Similarly, for the search strategy $\mathcal{T}$, Wiggins (2006) explains that another interpretation function is needed $\langle\langle .,.,. \rangle\rangle$ which, given three well-formed $\mathcal{R}$, $\mathcal{T}$ and $\mathcal{E}$ sets computes a function which maps two totally ordered subset of $\mathcal{U}$; c_{in}, and c_{out}. This function operates on members of $\mathcal{U}$ and not just on members of $\mathcal{C}$ because it is necessary to describe and simulate behaviours which are not completely well-behaved (Wiggins 2006). Therefore we get the expression: $c_{out} = \langle\langle \mathcal{R}, \mathcal{T}, \mathcal{E} \rangle\rangle (c_{in})$

Having different sets; $\mathcal{R}$ for the nature of the artefact, and $\mathcal{T}$ for the search strategy gives the possibility, explained by Wiggins (2006), to have transformational creativity by transforming $\mathcal{R}$ into $\mathcal{R}'$ or $\mathcal{T}$ into $\mathcal{T}'$ or both. This is an important feature because, for example, changing $\mathcal{R}$ is a way to change the constraints of the conceptual space, and it might be called transformational creativity in Boden (1990) terms and is equivalent to a paradigm shift. Changing $\mathcal{T}$ only affects the agent using that $\mathcal{T}$ (Wiggins 2006) but the agreed nature of an artefact remains the same.

Wiggins (2006) points out that in $\mathcal{C}$ there exist $\mathcal{C}_!$ and $\mathcal{C}_?$, concepts discovered and concepts not discovered yet respectively. Given $\mathcal{R}$ and $\mathcal{T}$ sets, some concepts in $\mathcal{C}_?$ may not be accessible, and even changing $\mathcal{R}$ (transformational creativity in Boden's terms), they might remain non accessible. By changing the search strategy $\mathcal{T}$ the elusive concepts in $\mathcal{C}_?$ might be accessible. This means that by transforming the search strategy one may find by exploration concepts $\mathcal{C}_?$ in $\mathcal{C}$. Boden (1990) suggests that transformational creativity is more significant that the explorational one. Wiggins (2006) explains that this formulation shows that this suggestion of Boden might not be true.

Wiggins (2006) explains that Boden's idea of transformational creativity is to change the rules that define the conceptual space. Wiggins defines two sets of rules, $\mathcal{R}$ and $\mathcal{T}$. Then the transformational creativity consists of changing either of them or both.

A syntax checker that selects $\mathcal{L}$ elements which are well formed is necessary. Therefore the transformations of $\mathcal{T}$ or $\mathcal{R}$ will be well formed in terms of any interpreter. Transformation means building new $\mathcal{L}$ subsets of the old ones (Wiggins 2006).

Wiggins (2006) explains that if we use a meta-language, $\mathcal{L}_\mathcal{L}$, for $\mathcal{L}$, which can describe the construction of new members of $\mathcal{L}$ from old ones, we can pair it with an appropriate interpreter, to allow us to search the space of possibilities. $\mathcal{L}_\mathcal{L}$ can be used to describe this task too. Then, we can evaluate the quality of transformational creativity, with some Ω function (Wiggins 2006). Then it could be possible to specify interpreters, $[\![.]\!]$ and $\langle\langle .,.,. \rangle\rangle$, which will interpret a rule set $\mathcal{T}_\mathcal{L}$ applied to an agenda of potential sequences in $\mathcal{L}$, such an interpreter could work for both $\mathcal{L}$ and $\mathcal{L}_\mathcal{L}$ (Wiggins 2006). Then, the evaluation function Ω, could be expressed as a set of sequences $\mathcal{E}_\mathcal{L}$ in $\mathcal{L}_\mathcal{L}$ and use $[\![.]\!]$ to execute it (Wiggins 2006). The transformational creativity system can now be expressed as an exploratory creative system working at the meta-level of representation (Wiggins 2006).

Wiggins suggests that, for true transformational creativity to take place the creator needs to be in some sense aware of the rules he/she/it is applying. This self-awareness, suggested by (Wiggins 2006), is what makes a creator able to formalise his/her/its own $\mathcal{R}$ and $\mathcal{T}$ in terms of the meta-language $\mathcal{L}_\mathcal{L}$. So without that self-awareness, a creator cannot exhibit transformational creativity (Wiggins 2006).

Wiggins points out that Boden's supposition that creative agents are well-behaved, in the sense that they either stick within their conceptual space, or alter it politely and deliberately by transformation may not be adequate. There are some situations in which agents may have a different behaviour which can be useful to analyse the system, they may also give information to switch to transformational creativity. They are grouped in (Wiggins 2006) into the terms *Uninspiration* and *Aberration*.

Uninspiration occurs in three different forms:

Hopeless uninspiration: there are not valued concepts in the universe.

Conceptual uninspiration: there are not valued concepts in the conceptual space.

Generative uninspiration: the search strategy of the creative agent does not allow it to find valued concepts

These categories are related to the value of the concepts. An agent can not even start working in the first situation. The second one requires redefining the constraints of the conceptual space. The third case indicates that the agent is not able, by the actual search strategy, to find valued concepts. A solution to this could be to modify the search strategy of the agent (Wiggins 2006).

Aberration is a situation where a creative agent is traversing its conceptual space. The strategy $\mathcal{T}$ enables it to create another concept which does not conform to the constraints required for membership of the existing conceptual space (Wiggins 2006).

Wiggins terms this aberration, since it is a deviation from the norm as expressed by $\mathcal{R}$. The choice of this rather negative terminology is deliberate, reflecting the hostility with which changes to accepted styles are often met in the artistic world (Wiggins 2006).

Aberrant concepts are very interesting because they are not part of $\mathcal{C}$ but the system might be able (by $\mathcal{T}$) to find concepts outside the constraints of the conceptual space defined by $\mathcal{R}$. The evaluation $\mathcal{E}$, of this concepts, has to be analysed carefully because, as expressed by (Wiggins 2006) and it was also noted by (León and Gervás 2010), $\mathcal{E}$ should be capable of scoring the results of $\mathcal{T}$ even when they fall outside the set defined by $\mathcal{R}$.

An Exploration of Engagement and Reflection under the CSF: An ER-CSF Model

Sharples (1996) explains that an episode of writing does not begin with a single goal, but with a set of external and internal constraints. It was shown in section "Writing as a creative design" that constraints can be the task, a collection of resources, aspects of the knowledge and experience of the writer, and a primary generator.

The constraints provide tacit knowledge to guide the writing process and the cycle E-R forms the cognitive motor of writing (Sharples 1996). Now we apply the Creative Systems Framework (Wiggins 2006) to the Engagement and Refection cycle (Sharples 1996)

Universe ($\mathcal{U}$) is a multidimensional space, whose dimensions are capable of representing anything, including the set of written materials or stories.

Language ($\mathcal{L}$) is a common language from which rules will be obtained.

Rules ($\mathcal{R}$) is formed with the set of constraints which form conceptual spaces. As explained by Sharples (1996) the operations a writer performs at each stage are different, so different results are produced at each stage. There will be $\mathcal{R}_E$ and $\mathcal{R}_R$ sets of rules to produce $\mathcal{C}_E$ and $\mathcal{C}_R$, conceptual spaces for Engagement and Reflection respectively.

$$\mathcal{R}_E \rightarrow \mathcal{C}_E \quad and \quad \mathcal{R}_R \rightarrow \mathcal{C}_R$$

Traversing strategy ($\mathcal{T}$) represents the strategy by which an agent produces an output in practical terms, they are the rules which define the way an agent will traverse $\mathcal{C}$. A writer can have different strategies to traverse the space. It was shown in section "Writing as a creative design", that a writer can have a strategy in which he is constantly reviewing the written material or, in other case, reviewing it after a long period of engaged writing. In any case, following Sharples, a writer produces written material through the strategy of an Engagement and Reflection cycle. As the output can be different due to switching frequency between Engagement and Reflection stages and because the operations performed, there are also two sub-strategies, $\mathcal{T}_E$ and $\mathcal{T}_R$:

1. $\mathcal{T}_E$ to traverse the space $\mathcal{C}_E$. When a writer is generating a chain of associated ideas and turning them into text.
2. $\mathcal{T}_R$ to traverse the space $\mathcal{C}_R$. When a writer is reviewing (and possibly making modifications), contemplating (exploring knowledge and transforming conceptual spaces) and planning for the next execution of engagement.

Evaluation ($\mathcal{E}$) It was outlined by Sharples (1996) that design problems are open-ended and can not be fully specified, they do not have a fixed set of goals or a sequence of steps, so, they cannot be evaluated in terms of their proximity to the goal. Sharples (1996) also highlights that a design task inevitably implies a subjective value judgement. Sharples (1996) explains that an engaged writer devotes full attention to creating a chain of associated ideas and converting them into text and nothing more can be done. Even if this is the case, at some point, a decision to switch to a reflective state is made and this might involve some kind of evaluation of the written material, for example, the extension of the material. During the reflective state, a writer reviews the material, contemplates it and makes plans, this involves the use of constraints, to get a set of criteria to evaluate the material. In the same way that there are two sets of rules $\mathcal{R}_E$ and $\mathcal{R}_R$ that define the conceptual spaces for the Engagement and Reflection stages, two sets can also be considered for the evaluation of concepts; $\mathcal{E}_E$ for Engagement and $\mathcal{E}_R$ for Reflection.

Concepts and rules

In a conceptual space $\mathcal{C}$, it is possible to find *concepts*. The proposal of Sharples (1996) does not indicate a particular type of concept to be found in a conceptual space other that written material. Sharples explains that, during Engagement a writer produces a chain of associated *ideas*. During Reflection the material generated in engagement is reviewed and, possibly, modified.

There are different types of constraints in this account to develop new written material, for example: knowledge and experience of the author, materials and resources, the task, etc.. Constraints have particular definitions but it can be said that there is a common language to define them. A convenient language of all constraints L_C could be represented by expression 1.

$$L_C = Language\_of\_Constraints \qquad (1)$$

Wiggins (2006) explains that $\mathcal{R}$ and $\mathcal{T}$ sets are needed to have the rules for the conceptual space and the strategy by which it will be traversed. In order to build those sets, we need a common language to define them too. $\mathcal{R}$ and $\mathcal{T}$ are defined by the set of constraints. We can use expression 1 to define a common language.

$$\mathcal{L} = L_C \qquad (2)$$

The set of rules $\mathcal{R}$, which defines $\mathcal{C}$, represent the agreed nature of what a concept is. $\mathcal{R}$ is a subset of $\mathcal{L}$ and can be described using (2). For this analysis, this account has two sets of rules; $\mathcal{R}_E$ and $\mathcal{R}_R$, for $\mathcal{C}_E$ and $\mathcal{C}_R$ conceptual spaces. The expression (3) can be produced.

$$\mathcal{R}_E \subset \mathcal{L}, \quad \mathcal{R}_R \subset \mathcal{L} \qquad (3)$$

By using an interpretation function $[\![.]\!]$, members of $\mathcal{U}$ which belongs to $\mathcal{C}_E$ and $\mathcal{C}_R$ conceptual spaces are chosen. We get the expression 4

$$\mathcal{C}_E = [\![\mathcal{R}_E]\!](\mathcal{U}), \quad \mathcal{C}_R = [\![\mathcal{R}_R]\!](\mathcal{U}) \qquad (4)$$

Sharples (1996) explains that, during Engagement there is no evaluation because the writer devotes full attention to generate the text and therefore it could be said that the set of evaluation rules $\mathcal{E}_E$, for concepts in $\mathcal{C}_E$, is empty. In contrast, during Reflection, there is an active evaluation ($\mathcal{E}_R$) of the written material. Expression (5) can be produced.

$$\mathcal{E}_E \subset \mathcal{L}, \quad \mathcal{E}_R \subset \mathcal{L} \tag{5}$$

There are also two strategies, $\mathcal{T}_E$ and $\mathcal{T}_R$ (Engagement and Reflection strategies respectively), useful to traverse $\mathcal{C}_E$ and $\mathcal{C}_R$ conceptual spaces. $\mathcal{T}$ is a subset of $\mathcal{L}$ and can be described using expression 2. Expression 6 is produced.

$$\mathcal{T}_E \subset \mathcal{L}, \quad \mathcal{T}_R \subset \mathcal{L} \tag{6}$$

Wiggins (2006) explains that an interpretation function $\langle\langle .,.,. \rangle\rangle$ is needed, which given three well-formed $\mathcal{R}$, $\mathcal{T}$ and $\mathcal{E}$ sets maps two totally ordered subset of $\mathcal{U}$; c_{in}, c_{out}. The interpretation function is one, but there are two different sets of rules constraining the conceptual space $\mathcal{R}_E$ and $\mathcal{R}_R$, two sets $\mathcal{T}_E$ and $\mathcal{T}_R$ for the Engagement and Reflection search strategies and two sets $\mathcal{E}_E$ and $\mathcal{E}_R$ for evaluation of concepts. So, given a c_{in} input subset of $\mathcal{U}$, it is possible to obtain outputs (subsets of $\mathcal{U}$).

$$\begin{aligned} c_{out\_Engagement} &= \langle\langle \mathcal{R}_E, \mathcal{T}_E, \mathcal{E}_E \rangle\rangle (c_{in}) \\ c_{out\_Reflection} &= \langle\langle \mathcal{R}_R, \mathcal{T}_R, \mathcal{E}_R \rangle\rangle (c_{in}) \end{aligned}$$

These functions can operate on members of $\mathcal{U}$ and not just on members of $\mathcal{C}_E$ or $\mathcal{C}_R$. They can describe and simulate behaviours which are not completely well-behaved as suggested by Wiggins (2006).

Aberration in ER-CSF

Wiggins (2006) proposes the term aberration for the situation when an agent is able to create by $\mathcal{T}$ another concept which does not conform to the constraints ($\mathcal{R}$) required for membership of the conceptual space. Sharples (1996) does not give a complete definition of the rules that comprise the conceptual space. In fact he explains that this set of rules depends on the writer and the particular constraints for a particular task. Sharples (1996) points out that, some writing displays such radical originality that we call it creative. Here, this "radical originality" could be a behaviour where the product does not conform to the constraints of the conceptual space.

Uninspiration in ER-CSF

For this account of writing, there is no specific definition of conceptual spaces. It depends on the writer to define a set of constraints to define the conceptual space, and also the strategy of the writer.

Sharples (1996) explains that, for example, the resources the writer uses; paper, pencil, etc., can affect the writing task. When there is a problem with one of the resources, that problem can block the writer if there is no alternative available. This could be an example of generative uninspiration explained by Wiggins (2006), where the strategy does not allow the writer to find valuable concepts in the conceptual space and needs to be changed.

Implementation examples

Example 1: MEXICA

MEXICA is an implementation of the computer model of creativity E-R proposed by Pérez y Pérez (1999).

The main goal of MEXICA is to produce novel and appropriate short stories as a result of an Engagement-Reflection cycle without the use of predefined story-structures which was built with many modifiable parameters to experiment with the process of creating a new story plot (Pérez y Pérez 1999).

MEXICA needs two inputs provided by the user: a set of Primitive Actions (PA) and a set of Previous Stories (PS).

MEXICA has a number of constraints, they will form the conceptual spaces and also define the strategies to build a story.

They are divided in the following categories:

Context Constraints are structures that represent the state of the current story (Pérez y Pérez 1999).

Knowledge Constraints are constituted by the experience, knowledge and beliefs of the writer.

Guidelines constrain the material to satisfy the requirements of novelty and interest (Pérez y Pérez 1999).

General constraints include rhetorical and content constraints not included in the previous classifications. They are formed by a set of requirements that must be satisfied by all events retrieved from memory and are necessary for MEXICA to operate correctly (Pérez y Pérez 1999).

In MEXICA a story is a sequence of events or actions which are coherent and interesting. An action has preconditions and post-conditions, useful to give coherence to a story and to know the consequences of the execution of an action respectively.

When an action is executed, consequences arise and they generate a story context. Story contexts are useful in MEXICA because they linked an action with the next one.

Having an action linked to the next is not enough. In MEXICA it is also needed to link an action with the previous one in order to guarantee coherence, this is how preconditions are taken into account. In MEXICA a coherent sequence is that where all preconditions of all actions are satisfied. Here we have an important concept in MEXICA: *coherence*. Coherence is a property of stories and they can only be coherent or non-coherent at a time.

Engagement in MEXICA During Engagement a sequence of actions linked by story contexts is produced. MEXICA retrieves possible next actions from memory using story contexts. Engagement selects one of the actions to continue the story appending it to the story in progress (Pérez y Pérez 1999).

During Engagement MEXICA does not verify if the story actions satisfy pre-conditions, so sequences of actions with unsatisfied pre-conditions might be produced (potentially non-coherent stories).

Reflection in MEXICA In contrast with Engagement, Reflection verifies pre-conditions for each action in the story in progress in order to produce a coherent story. When unfulfilled pre-conditions are detected in the story in progress, MEXICA fetches an action whose post-conditions satisfy such unfulfilled pre-conditions and inserts it. The process is repeated if new actions have unsatisfied preconditions (Pérez y Pérez 1999) During Reflection, only coherent stories can be produced.

MEXICA also implements heuristics to test if the story in progress is interesting. MEXICA assumes that the stories in the set of PS supplied are interesting and so its Tensional Representation is a good example to follow (Pérez y Pérez 1999).

Boden (1990) suggests that novelty is one important characteristic of creative acts. Novelty is also considered in MEXICA and during Reflection, there are rules to assess novelty. MEXICA verifies if the material produced during the Engaged state resembles too much any of the tales in the set of PS (Pérez y Pérez 1999) and if this is the case MEXICA changes the search strategy.

Example 2: Dev E-R

Dev E-R (Aguilar and Pérez Pérez 2015) (Developmental Engagement-Reflection) is a computational model that, inspired by Piaget's theory, simulates the assimilation-accommodation adaptation process. It is implemented with the computer model of creativity Engagement-Reflection. This model simulates adaptation as a creative activity.

In Dev E-R, a development agent is implemented to simulate the adaptation processes to a particular environment. The agent is initialised with basic knowledge structures called schemas, which represent innate behaviours observed in newborns. It is also capable of creating new knowledge structures as a consequence of its interaction with the environment (Aguilar and Pérez Pérez 2015). The objects with which it interacts have a number of characteristics the agent can sense (Aguilar and Pérez Pérez 2015).

When the agent begins to operate it sees objects as static or in-motion luminous spots which have a position within the field of vision. The spots detected are used to create an internal representation of what the agent sees. This representation is called the current context (Aguilar and Pérez Pérez 2015).

At the beginning the agent can not recognise all the visual characteristics of objects, contexts can only describe bright spots appearing, moving and disappearing. These contexts are then used to build schemes. Eventually, through interaction with their environment, the agent acquires the ability to see spots not only as luminous things, but as visual elements with different colours and sizes. Whenever an object enters the field of view of the agent, the values of the variables representing the characteristics of the object increases in one. When the value of the variable associated with any of the differentiated colours or sizes reaches a certain predefined value N, then it is said that such a characteristic has sufficient stimulation and the agent acquires the ability to recognise it and use it to construct its knowledge structures (Aguilar and Pérez Pérez 2015).

The current context is a structure composed of 3 parts: (1) the characteristics of the object that is in the centre of attention of the agent (colour, size, movement and position), (2) the affective responses, emotional states and motivations triggered by such an object, and (3) current expectations of the agent (Aguilar and Pérez Pérez 2015).

Dev E-R schemes are knowledge structures that simulate the sensorimotor schemes, which is a psychological construction that gathers together the perceptions and associated actions involved in the performance of a behaviour. It includes knowledge about the context in which the behaviour was performed, as well as expectations about its effects (Aguilar and Pérez Pérez 2015).

The agent has adaptation mechanisms to simulate assimilation, accommodation and cognitive equilibration processes. They represent its core component, since they allow it to develop cognitively through interaction with the virtual world. This is done either by modifying its perception of the environment so that it fits the current knowledge (adaptation by assimilation) or by modifying and producing new knowledge when it does not match reality (adaptation by accommodation). This model simulates adaptation as a creative activity (Aguilar and Pérez Pérez 2015).

The Dev E-R model has two ways of using and building knowledge of the agent: (1) automatically, through Engagement, and (2) analytically through Reflection (Aguilar and Pérez Pérez 2015).

Engagement in Dev E-R Engagement takes the current context and use it as a cue to probe memory in order to match a scheme that represents a situation similar to the current one. If the current context matches more than one scheme, the system selects only one of them. When a scheme is matched, the agent executes the associated action. Then, the agent perceives its world again, updates the current context and the cycle continues If the agent can not associate any schema an impasse is declared. In this case, it switches to Reflection (Aguilar and Pérez Pérez 2015).

Reflection in Dev E-R During Reflection, the agent tries to analyse the current situation and, with the help of some pre-defined strategies, tries to deal with the unknown situations (Aguilar and Pérez Pérez 2015). In Dev E-R, accommodation implies the creation of new schemes and the modification of existing ones as a result of dealing with unfamiliar situations (Aguilar and Pérez Pérez 2015). The creation and modification of the schemes is carried out by means of the following methods: generalisation or differentiation. The process of generalisation takes place in two situations: (1) when the agent recovers an object of interest by chance and then it generalises that sole experience in an abstract schema; and (2) when the agent detects that the same action can recover various objects with different features and then it generalises this knowledge in an sole schema (Aguilar and Pérez Pérez 2015).

As a result of development of the agent, the search mechanisms during Engagement change to adapt to the increased number of experiences.

MEXICA and Dev E-R under the ER-CSF

Two implementations of the Engagement-Reflection model have been presented. Now they are analysed based on the ER-CSF model presented in section "An ER-CSF Model".

Sharples (1996) explains that a writing process depends on a set of external and internal constraints that will guide the process. The Engagement and Reflection cycle forms the cognitive motor of this process considering that set of constraints.

Also, there are some concepts presented in the ER-CSF model that should now be related to the implementations examples.

Universe ($\mathcal{U}$) is the multidimensional space, whose dimensions are capable of representing anything. for Sharples (1996) the set of written materials is the important one. In MEXICA (Pérez y Pérez 1999), is the set of short stories about the Mexicas. In Dev E-R (Aguilar and Pérez Pérez 2015) is the set of behaviours of the agent.

Language ($\mathcal{L}$) is a common language from which rules will be obtained. This language changes for each case because they are not in the same domain. Sharples' (1996) account and MEXICA could be more related because they generate written materials but they do not consider the same kind of constraints, so the language is different.

Rules ($\mathcal{R}$) is formed with the set of constraints which build conceptual spaces. For Sharples (1996), the operations a writer performs in Engagement and Reflection are different, and so are the results produced at each stage. In MEXICA, there are rules for the operation of the system which form conceptual spaces of coherent or non-coherent stories. Also, some constraints are useful to an specific stage. In Dev E-R the agent deals with a set of constraints which include; objects, properties of those objects, expectations, actions the agent can perform, a number of initial basic behaviours which will be developed, etc. Engagement deals with the interaction with the context related to familiar schemas (behaviours) reinforcing the stimuli of a property or the pleasure over a particular situation. Reflection deals with unfamiliar situations from which the system will create new schemas or synthesise them.

In any case, the set of constraints which define conceptual spaces are different for Engagement and Reflection and also the processes are different in each stage and therefore they produce different results. So, following the ER-CSF model, there are $\mathcal{R}_E$ and $\mathcal{R}_R$ sets of rules to produce two different conceptual spaces $\mathcal{C}_E$ and $\mathcal{C}_R$, for Engagement and Reflection respectively.

Traversing strategy ($\mathcal{T}$) represents the strategy by which an agent produces an output in practical terms. For Sharples (1996), a writer can have different strategies due to knowledge but also materials. In MEXICA (Pérez y Pérez 1999), the previous knowledge is important but not the material or resources. In contrast, Dev E-R (Aguilar and Pérez Pérez 2015) uses a representation of objects which are the surrounding context of the agent and they may represent its resources or materials, the previous knowledge is also important but at the beginning is very limited, as the process goes on it is incremented. $\mathcal{T}$ are the rules which define the way an agent will traverse $\mathcal{C}$.

MEXICA produces a story and Dev E-R produces the knowledge of the agent, both through the strategy of an Engagement and Reflection cycle, as explained in section "Writing as a creative design". The outcomes of each stage can be different because they do not perform the same operations to continue a story in progress. So, there are two sub-strategies, $\mathcal{T}_E$ and $\mathcal{T}_R$.

1. $\mathcal{T}_E$ to traverse the space $\mathcal{C}_E$. For MEXICA, when the system is working in the Engagement state and when actions are being appended using story contexts and no pre-conditions of any action are verified and when the agent faces familiar situations and updates existing schemas for Dev E-R.

2. $\mathcal{T}_R$ to traverse the space $\mathcal{C}_R$. For MEXICA, when, in order to produce a coherent story, pre-conditions are verified and when Engagement is not able to retrieve actions from memory to continue the story in progress and an impasse is declared. And for Dev E-R when the agent faces unfamiliar situations and it needs to adapt to that situation by creating schemas or synthesising them.

Evaluation ($\mathcal{E}$) It was explained by Sharples (1996) that no specific goals are established and that there is no specific evaluation function of the material because the design task is an open-ended problem. But he also explains that at some point some evaluation should be made. MEXICA does not have evaluation rules during Engagement but it has rules to evaluate novelty and interest implemented in Reflection. Depending on the result of the evaluation, guidelines might be updated and the strategy $\mathcal{T}_E$ might change (this may be seen as strategy-transformation, $\mathcal{T}$-transformational creativity in Wiggins' (2006) terms). Dev E-R has rules to evaluate novelty and adaptation of the agent to the virtual world, but in contrast with the ER-CSF model, this does not happen in a particular stage. We can say that both stages have the same set of rules for evaluation. MEXICA in the original model and Dev E-R do not have specific goals to evaluate the outputs, but Pérez y Pérez (2015) introduces a new model for MEXICA for evaluating its outputs. In this new model two instances of MEXICA work together to produce a story, they have different sets of PS but when the story is finished they incorporate the new story to their knowledge structures, so, they change, producing more structures or widening the existing ones, This is a relevant characteristic and will be analysed in a future work but for this analysis, we will continue using the original model in order to have a solid starting point.

As explained in section "Writing as a creative design", there could be evaluation during each stage, so, two sets can also be considered for the evaluation of concepts; $\mathcal{E}_E$ for Engagement and $\mathcal{E}_R$ for Reflection.

Concepts and conceptual spaces are different for this examples, but they all need the rules to build the conceptual spaces from their particular definition of a concept. A common language is needed to define the rules of this model. From this language we can be able to define the conceptual space(s), strategies and evaluation rules.

From section "An ER-CSF Model" we can use expression 1 to produce a common language even if the constraints are different for this examples. We also can generate the sets of rules $\mathcal{R}_E$ and $\mathcal{R}_R$ with the expression 3.

With the sets of rules $\mathcal{R}_E$ and $\mathcal{R}_R$ defined, we can use an interpretation function to chose members of $\mathcal{U}$ which belongs to $\mathcal{C}_E$ and $\mathcal{C}_R$ conceptual spaces as it was shown in expression 4.

Following the description in section "An ER-CSF

Model", we can produce sets of rules for evaluation. Sharples (1996) explains that, during Engagement there is no evaluation because the writer devotes full attention to generate the text and therefore it could be said that the set of evaluation rules for Engagement $\mathcal{E}_E$, for concepts in $\mathcal{C}_E$, is empty and the same can be said about MEXICA, so we get $\mathcal{E}_E = \varnothing$. These are special cases we can get from expression 5 in section "An ER-CSF Model". And finally we can get search strategies for both systems using expression 6.

Aberration Pérez y Pérez (1999) explains that in MEXICA a story is a sequence of actions, but it is also important that the sequences of actions are logical and coherent. A logic and coherent sequence of actions is that where the preconditions of all actions in the sequence are satisfied (Pérez y Pérez 1999).

In Engagement there is no guarantee to produce a coherent story. Furthermore, when Engagement receives a coherent story from Reflection, it appends a new action to the story and that operation can modify the story producing a potentially non-coherent story. When a non-coherent story is generated, that story does not conform to the constraint of $\mathcal{R}_R$ and is, therefore, an aberrant concept for Reflection.

What is important to notice here is that as part of the MEXICA process the system is exploring options out of the scope of the main objective of MEXICA (out of the scope of $\mathcal{C}_R$ too, therefore aberrant concepts) which is to produce coherent stories.

In Dev E-R the concepts that can be generated can only refer to the virtual world in which the agent is developing, the objects in the world and their characteristics, and also to the action the agent can perform. There are many variations but all concepts that can be generated meet the rules of the conceptual space, so it is not possible to find aberrant concepts.

Uninspiration In MEXICA and Dev E-R, when Engagement is not able to find actions or schemas related to the current context an impasse is declared. The search strategy $\mathcal{T}_E$ is not being able to create a new concept. This can be seen as uninspiration in Wiggins' (Wiggins 2006) terms. When the uninspiration is due to the generative process it can be fixed by changing the strategy $\mathcal{T}$. MEXICA and Dev E-R can break an impasse by switching to Reflection and they both change the current context adding a new action or applying a change in the state of the agent (e.g. to move its head) or its knowledge.

Once Reflection has changed the context, the system switches to Engagement but now it can be considered that strategy $\mathcal{T}_E$ has changed because different knowledge constraints and, guidelines for MEXICA, are available.

This implementation examples are basically the same in terms of the model but have differences regarding the evaluation rules, specifically $\mathcal{E}_E$. MEXICA has an empty set of evaluation rules for engagement (the same as Sharples' (1996) account), but Dev E-R has the same set of rules for Engagement than for Reflection $\mathcal{E}_E = \mathcal{E}_R$. They are also different because Dev E-R can not generate aberrant concepts and MEXICA can.

Conclusions

Sharples' (1996) proposal, in which it is explained that, a writing process is guided by constraints and by an Engagement and Reflection cycle has been analysed using the Creative Systems Framework (CSF: Wiggins 2006) to achieve a better understanding of this model and to better apply it in computer systems.

Two examples have been shown that implement in a computational system an Engagement and Reflection cycle that guided by a set of constraints produce a result. The examples shown have been analysed based on the review of Sharples' proposal under the CSF. We conclude that the model product of this analysis shows particular characteristics of operation for each stage of the cycle and therefore the results are not always the same. Also, the constraints are used differently in each stage and that affects the results.

This analysis shows a clear differentiation between the conceptual spaces, rules that define them, strategies and evaluation for Engagement and Reflection. This differentiation makes an important contribution in the systems since it allows the system to explore conceptual spaces whose members may not belong to its conceptual space. It is also important to notice how one stage can modify the way the other operates, changing the constraints.

Acknowledgments

We would like to thank the National Council of Science and Technology (CONACYT) in México for its sponsorship.

References

Aguilar, W., and Pérez Pérez, R. 2015. Dev E-R: A computational model of early cognitive development as a creative process. *COGNITIVE SYSTEMS RESEARCH* 33:17–41.

Boden, M. A. 1990. *The Creative Mind: Myths and Mechanisms*. Abacus.

Colton, S. 2008. Creativity Versus the Perception of Creativity in Computational Systems. In *Proceedings of the AAAI Spring Symposium on Creative Systems*, volume 8, 14–20.

León, C., and Gervás, P. 2010. The role of evaluation-driven rejection in the successful exploration of a conceptual space of stories. *Minds and Machines*.

Pérez y Pérez, R. 1999. *MEXICA : A Computer Model of Creativity in Writing*. Ph.D. Dissertation, The University of Sussex.

Pérez y Pérez, R. 2015. A computer-based model for collaborative narrative generation. *Cognitive Systems Research* 36-37:30–48.

Ritchie, G. 2007. Some empirical criteria for attributing creativity to a computer program. *Minds and Machines* 17(1):67–99.

Sharples, M. 1996. An account of writing as creative design. *The science of writing: Theories, methods, individual differences, and applications.* (January):127–148.

Wiggins, G. A. 2006. A preliminary framework for description, analysis and comparison of creative systems. *Knowledge-Based Systems* 19(7):449–458.

Surprise Walks: Encouraging users towards novel concepts with sequential suggestions

Kazjon Grace[1], Mary Lou Maher[2], Nicholas Davis[2], and Omar Eltayeby[2]
[1]The University of Sydney, and [2]UNC Charlotte
[1]Australia, and [2]USA
kazjon.grace@sydney.edu.au, {m.maher, ndavis64, oeltayeb}@uncc.edu

Abstract

In this paper we present a proof-of-concept of how co-creative systems could guide their users to appreciate artefacts that are currently too novel. Given that too-novel artefacts are off-putting, and domain experience reduces novelty, this situation will arise often when a co-creative system has more domain experience than its user. We present some experiments demonstrating a strategy for generating sequences of concepts to present to users. These sequences are designed to provide the necessary background to allow users to appreciate a highly-novel "target" artefact. Our strategy is based on generating and then traversing "surprise space", a form of conceptual space in which concepts which are surprising in the same contexts are proximal. We implement this strategy, which we call a "surprise walk", in the domain of recipes using a word embedding algorithm with a modified objective function that co-locates features that are similarly surprising.

Introduction

Consider the case where a human and computer are collaborating on a creative task (aka "co-creativity"), but the latter knows more than the former. Where we are today, at the very beginning of usable co-creative systems, that might seem like an edge case. We contend, however, that in time it might describe the majority of such interactions. Imagine a future in which co-creative systems are commonplace: it is likely that the majority of their users will not be experts. It follows that co-creative systems will often possess knowledge their users do not, even discounting situations in which they are explicitly being used for education.

This creates a challenge for systems that generate content more novel than their users are currently prepared to accept. Under the Wundt curve model (Berlyne 1966; Saunders and Gero 2001), there is a peak level of novelty at which positive affective response is maximised. Either side of that peak the response becomes negative: either too boring (insufficient novelty) or too alien (overwhelming novelty). Creative systems operating with more knowledge than their users will often generate artefacts that are desirably novel to the system, but (if we accept the Wundt curve model) overwhelmingly so to their users. Greater knowledge would lead to more accurate expectations, and thus less surprise.

If the human has decision-making power in the creative task, as is common in co-creative systems, then a co-creative system must convince its users of the benefits of its creations. How could a co-creative system "guide" a human towards a creative (i.e. novel and valuable, (Newell, Shaw, and Simon 1959)) region of the space of possible artefacts, even if those artefacts were currently overwhelmingly novel to the user? One answer is for systems to seek "user appropriate" rather than maximal novelty. Another is persuasion.

In this paper we explore how computationally creative systems might persuade humans to appreciate more novel artefacts. We propose "surprise walks", a strategy for generating sequences of increasingly surprising concepts. These sequences start with a goal concept that the system desires the user be able to appreciate. The strategy is then to work backwards, decreasing the level of surprise, until a concept that the user can appreciate is reached. A creative system could then expose its user to artefacts exhibiting each concept in turn. Where necessary, multiple artefacts exhibiting a concept could be presented until the user appears to have comprehended or accepted it. The intent is to pique user curiosity over time, and maintain that curiosity state while working towards a goal (Grace and Maher 2015). That the user is being taken for a "surprise walk" may or may not be communicated to the user, which raises a variety of ethical issues which we return to in the discussion.

We present a model of the surprise walk process and additionally introduce the concept of "surprise space" on which the process is based. A "surprise space" is a specialised kind of conceptual space in which proximal concepts are *similarly surprising*, rather than being literally similar themselves. We also present a prototype implementation of a surprise walk generator, capable of accepting a target surprise and a simple artificial user profile and outputting sequences of concepts. We present and discuss the results of this prototype, comparing the sequences that can be generated using a surprise space to those generated by the same process using a conceptual space based on literal similarity.

Background

This research occurs at the intersection of two literatures: co-creativity and computational models of surprise and curiosity. To date, most research in co-creative systems has not explicitly considered the idea of imbuing such systems with the desire to spark curiosity. Similarly, most research in computational curiosity has not considered the context of co-creativity.

Co-creative systems

A variety of co-creative systems are able to influence their user's behaviour. The Drawing Apprentice (DA) is a co-creative drawing partner that collaborates with users on a shared drawing (Davis et al. 2016; 2018). The system analyses the user's input and responds with complementary objects to inspire the user's creativity and sustain engagement over time. The Sentient Sketchbook is a co-creative game level design tool that leverages user input to generate design alternatives that may surprise the user and support their creativity (Liapis, Yannakakis, and Togelius 2013). Clark et al. (2018) describe a machine-in-the-loop writing system that provides surprising and unpredictable output designed to inspire user creativity. Similar systems include Creative Help (Roemmele and Gordon 2015) and Say Anything (Swanson and Gordon 2012).

In none of the above systems is there a capacity to reason beyond the next step in designing the current artefact. That is not an criticism, doing so is simply out of their scope. They assist human creators by providing in-the-moment suggestions. This research explores a way for co-creative systems to form longer-term goals.

Computational surprise

The concepts of novelty, unexpectedness and surprise have been the subject of many definitions in the computational creativity, artificial intelligence and cognitive science literatures. For the purposes of this study we define novelty as the degree to which an artefact differs from those that have come before within that creative domain. There are many ways to operationalise that definition, but building on our previous work we argue that the best way to do so is by quantifying the expectations of the agents acting in that domain, and then measuring the degree to which those expectations are violated by an artefact (Grace and Maher 2014). We call this an *unexpectedness* based approach to novelty. Similar approaches have been adopted by Macedo and Cardoso 2001 and Gravina et al 2016.

In most of our work expectations are defined in terms of sets of features that co-occur regularly, with unexpected artefacts being those which exhibit sets of features that co-occur only infrequently. "Surprise" is an agent's response to unexpectedness, although in most contexts this can be used interchangeably with unexpectedness. We measure the amount of surprise using the negative base-2 log of ratio of the co-occurrence probability of those features to their probability of them occurring separately. "Surprise walks" are thus an exploration of how a co-creative system could expose an individual to a sequence of surprising artefacts, each not only similar but more unlikely than the last.

Computational curiosity

Berlyne (1966) describes the prevailing psychological theories of curiosity as curiosity-as-state and curiosity-as-trait. Curiosity-as-trait refers to an innate ability of a person, and individuals differ in how much curiosity they have. Curiosity-as-state refers to a motivational state of a person that causes the person to seek novel stimuli, and it varies within each person according to their context. Curiosity-as-state is malleable: curiosity can be encouraged by external events or contexts. A computational model of state curiosity is one that seeks surprising events or objects and in co-creativity a computational model of surprise can present stimuli that encourages user curiosity. Curiosity-as-state has been integrated into cognitive systems in the past, such as Saunders and Gero (2001) and Merrick and Maher (2009) .

Berlyne additionally proposed that state curiosity can be considered along two dimensions: epistemic vs perceptual, and diversive vs specific. In the first dimension, perceptual curiosity is the drive towards novel sensory stimuli and epistemic curiosity is the drive to acquire new knowledge. Our surprise walks could theoretically be applied in either case, but we are exploring epistemic applications. Along the second dimension, Berlyne describes diversive curiosity as unguided search for any new information or stimuli and specific curiosity is search for a novel solution to a specific problem or goal. The majority of current models of computational curiosity are diversive in nature, such as Saunders and Gero and Merrick and Maher mentioned above as well as Schmidhuber (2010). Our surprise walks are adopting the concept of specific curiosity: how a system could influence a user towards a novel goal.

Surprise Spaces

A conceptual space in a creative domain captures the ordering principles or underlying structure of that domain's concepts. In some conceptual spaces the artefacts are described in terms of dimensions that are meaningful to that domain (Gärdenfors 2004). If a conceptual space is constructed in this manner then the "concepts" are dimensions, and each point represents a hypothetical artefact. In other approaches globally meaningful dimensions are not required, instead proximal concepts are always similar (Boden 1996). In this approach each point in space represents a concept, and an artefact is composed from one or more concepts.

A surprise space is a particular kind of conceptual space in the latter tradition: proximity implies similarity. However the concepts that are distributed through that space are combinations of artefact features, each of which is assigned a surprise based on measures described in our previous work (Grace et al. 2017). As a simple example, consider a surprise space in the domain of recipes in which each point represents the combination of two ingredients. Some of those combinations (such as onion and garlic) will be of low surprise, while others (such as chocolate and garlic) will not. A surprise space need not be constructed of these simple unordered pairwise combinations of features, but could instead contain any combination of two or more elements that is meaningful to the domain: consecutive phrases of music, visual features combined with a particular caption word, or triplets of named entities appearing together in news articles. A surprise space is intended to augment, rather than replace any other form of conceptual space in a creative system's reasoning. We do not suggest that this way of constructing conceptual spaces is in any way superior to any other – it is simply different.

The principle of organisation in surprise space is the similarity between surprises. By carefully traversing this space we could construct a sequence of surprises that are increasingly but also similarly surprising. That sequence could transport a user from the borders of their current knowledge to some as-yet-too-alien combination. This journey through surprise space, which we call a "surprise walk", leverages the unique structure of a surprise space as a metacognitive aid. It guides a creative system's behaviour as a means for influencing the behaviour of its human collaborators.

But what does it mean for two surprises to be similar? There are several possible approaches here, and we prototype two in the proof-of-concept detailed below. The simplest is to average the similarity of each feature (or set of features) in the combination, using the best possible mapping between features to do so. For example, assume that each point in surprise-space represents a combination of two or more ingredients in a recipe. Given (A,B) and (C,D) as two such combinations, we can take the similarity between the two surprises as being:

$$max(s(A,C) + s(B,D)),(s(A,D) + s(B,C)))/2$$

Where $s(x, y)$ as a similarity measure for features x and y. Note, again, that this is specifically the similarity between two *surprises*, (A, B) and (C, D).

An alternative approach would be to construct a similarity measure between surprises. This is akin to comparing between two differences: how similar is the difference between A and B to the difference between C and D, to take the example above? In our recipe example this could be measured using a physiological model of taste, a molecular gastronomical model of chemical compounds, or the co-occurrence of ingredients. We introduce a hypothetically domain-independent approach below that performs this kind of surprise-to-surprise comparison based on whether the ingredients are *surprising in similar contexts*. Let's say A is soy sauce, B is chocolate, C is mushrooms and D is icing sugar (confectioner's sugar in North American English). Is the way soy sauce differs from chocolate similar to the way mushrooms differ from icing sugar? Our prototype says yes: soy sauce is surprising when combined with a similar list of things as mushrooms are, and the same with chocolate and icing sugar. For example, both soy sauce and mushrooms are surprising in combination with vanilla, apples, and barbecue sauce. Similarly, chocolate and icing sugar are both surprising in combination with steak, black pepper, and tofu.

In our proof-of-concept implementation we have implemented both approaches: literal feature similarity comparison as well as comparing the similarity of surprises directly.

Surprise Walks: Navigating surprise spaces

Our motivation in conceiving of "surprise walks" is to explore how co-creative systems could encourage their users towards appreciating concepts that they could currently consider too novel. We define a surprise walk as a sequence of combinations in a surprise space that a) are of monotonically increasing surprise, b) are sequentially proximal in the space, c) start with a combination familiar to the user and d) end with a target combination of (currently) overwhelming novelty. That target combination is not only novel, but so novel that the user cannot or will not appreciate it: it is off the right shoulder of their personal Wundt curve.

Additional constraints on the sequences might be desirable, such as ensuring that adjacent elements are not too dissimilar in their surprise ratings. The intent is that these sequences act as a long-term plan for the behaviour of the co-creative system. They could allow it to curate the new experiences of their human user and thereby influence that user's Wundt curve until the target combination is no longer overwhelming. This definition is sufficiently broad to permit a large variety of approaches to sequence generation. We describe one such approach below in the domain of recipes.

s-GloVe: A prototype surprise space

Our prototype surprise walk system is based on a word embedding algorithm called "GloVe" (Pennington, Socher, and Manning 2014), used for representing each word in a corpus of documents as a vector of numbers. We call our surprise-based modification of it "s-GloVe". Word embedding algorithms map each word that occurs in a corpus of documents (typically one in which each document is represented as a bag of words, i.e. a count of all words that occur, ignoring word order) into an abstract continuous space. This space typically has a few hundred dimensions. We selected GloVe for this work as it approximates the matrix of co-occurrences between features, a desirable quality in a model of unexpectedness. We treat each ingredient as a "word".

Representing each word as a continuous vector allows for capturing similarity between words: similar words are proximal in vector-space. The most common approach to measuring the "similarity" between two words is based on the concept of *distributional semantics*, or the idea that you can "know a word by the company it keeps" (Firth 1957). More precisely, distributional semantics states that similar words have similar distributions over what other words are likely to occur nearby (Harris 1954): they occur in similar sentences. Constructing a word embedding model such that words with similar contexts occur nearby in vector space makes all kinds of similarity-based tasks easier, including clustering, thematic analysis, document classification, and augmenting the training of other machine learning algorithms.

GloVe has become a standard for word embeddings as it is simple, scalable and robust. It operates by learning a vector of arbitrary numbers for each word in the corpus. Its objective is to construct those vectors such that the vectors of any two words can be used (via a mathematical transformation) to calculate how those co-occur. What exactly it means for two words to "co-occur" is dataset specific: it could be that they are both within the same sentence in a news article, within the same line in a poem, or within the same section in a scientific paper. The result is that the vector representation for each word encodes how that word correlates with every other word. When these word vectors are interpreted as points in space, nearby words co-occur with all other words in similar ways. Or, to put it another way, they share distributional semantics. GloVe uses gradient descent to construct its vectors, with the objective of minimising the difference between the true co-occurrence of words and the one reconstructed from the word vectors.

A full description of the GloVe algorithm can be found in the original paper, but two points are relevant to how we modified the algorithm to discover similar surprises. Firstly, GloVe's vectors exhibit locally linear relationships between words that capture their meanings. This means that the differences between similar pairs of words are themselves similar. The difference between woman and man (the subtraction of those two vectors) is similar to the difference between queen and king, or between aunt and uncle. Similarly, the differences between US cities and their zip codes are all similar, as are the differences between Fortune 500 companies and their CEOs. This extends also to grammatical structure, with the difference between comparative and superlative forms of the same adjective (e.g. "softer" vs "softest") being highly similar. We exploit this property in our prototype.

Secondly, to speed up training and produce more robust vectors the original GloVe algorithm lowers the impact of reconstruction error for rare words using a weighting function. It is by replacing this weighting function that we re-imagine GloVe as a space of similar surprises.

s-Glove: The distributional semantics of surprise

GloVe captures the meaning of words by quantifying the company they keep. S-GloVe captures the way words are surprising by quantifying the company they don't, or rather, the company in which they are unexpected. This is still a kind of distributional semantics, as it defines words by the statistical properties of their context. In practice, however, it leverages almost the opposite information to the basic GloVe approach. S-GloVe encodes the co-occurrence between only those word-pairs that are surprising. In doing so it effectively disregards all the commonly co-occurring words, which are the key information leveraged in the majority of distributional models. This creates a space where nearby surprises are similar because of why they are surprising, not because of what they are. This could permit a system to reason about *why* a particular combination is surprising/novel.

The GloVe cost function includes a weighting against rare words (Pennington, Socher, and Manning 2014). For the technical details consult the original paper, but in short it reduces the impact of the error in reconstructing the co-occurrence between words if that co-occurrence is low. The effect of this is that rare co-occurrences are not encoded as strongly in the word vectors, and do not affect word-word similarity as much. We replace this with a function that reduces the impact of co-occurrences which are unsurprising.

We use a test for statistical significance – the one-tailed version of Fisher's exact test – to quantify the evidence for whether a pair of words occur less frequently together than one would expect were they independent. This test draws from the word occurrence and co-occurrence data, and provides a p-value for the chance that they are actually significantly surprising. A sufficiently low p-value lets us reject the null hypothesis that this pair of words is not surprising (i.e. either independent or actually more likely to occur together). The specific weighting function we use in s-GloVe, which replaces $f(X_{ij})$ in Pennington et al (2014), is:

$$f(X_{ij}) = 1 - min(p_{ij}, \alpha) \tag{1}$$

where $p_{i,j}$ is the p-value of the left tail of the Fisher test for independence of words i and j, and α is a parameter controlling how small the impact of unsurprising word-pairs will be on the word vectors. As α approaches 0 unsurprising word pairs have effectively no impact as p-values above 0.999 are common. We used $\alpha = 0.1$ in our tests after some experimentation, as with higher values the s-GloVe space began to more strongly resemble the original GloVe space.

Dataset Description & Preprocessing Pipeline

We began with the Now You're Cooking dataset[1], as used in Kiddon et al 2016. The dataset contains around 80,000 unique recipes that have been shared on the Internet since the 1990's. The recipes are provided with their names, ingredients, quantities, units, preparation steps and tags. In our experiments we use the ingredient set and cuisine tags only, discarding for now the titles, quantities and preparation steps. We treated each ingredient, post-processing, as a single feature in our model, such as "white wine" or "parmesan cheese".

We used the New York Times' ingredient-phrase tagger[2] to extract from strings like "1/3 cup freshly shredded lettuce" the name of the ingredient itself (here "lettuce"). Manual cleaning of about 10% of recipes was required after this step, presumably due to differences between the NYT tagger's training data and our dataset. We also combined a number of less common ingredients (e.g. varieties of soy sauce or orange liqueur) into single categories for the purpose of simplicity. After parsing, cleaning, duplicate elimination, and deleting those recipes with less than three ingredients we ended up with 73,000 recipes. Figure 1 presents an example of our pre-processing, transforming complex ingredient strings into simple, corpus-coherent ingredient features.

Results: ingredient-ingredient similarity

To validate our ideas about what proximity in s-GloVe space represents, we compare the most similar ingredients to a target ingredient, i.e. the nearest neighbours in the vector space. We use cosine similarity in each case, and compare the same six ingredients between GloVe and s-Glove. In both cases (and throughout this paper) the ingredient vectors have 64 dimensions. In the case of the original GloVe paper the $xmax$ parameter is set to 0 to prevent de-emphasising rare words. We arbitrarily selected five highly dissimilar words as test cases: pine nuts (occurs in 485 recipes), cucumbers (1007 recipes), cayenne powder (1714 recipes),

[1] https://github.com/uwnlp/neural-checklist

[2] https://github.com/NYTimes/ingredient-phrase-tagger, as discussed at https://open.blogs.nytimes.com/2015/04/09/extracting-structured-data-from-recipes-using-conditional-random-fields

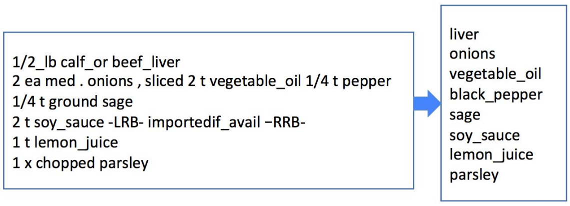


Figure 1: A set of "raw" recipe ingredients and the cleaned list used by our system, based on already-partially-preprocessed data by Kiddon et al (2016).

lentils (400 recipes), and apples (2270 recipes). The results are shown in Tables 1 through 5.

At the system's current level of development it is not yet feasible to objectively compare, via quantitative metrics or user feedback, the results of the s-GloVe surprise works to those of GloVe. Some interpretation must be permitted to judge the relative strengths and potential of the approaches. The results presented here are thus for the reader's own subjective digestion, although we believe they represent sufficient promise to continue investigating.

Table 1: Most similar to *"pine nuts"*.

Word (GloVe)	cosine	**Word (s-GloVe)**	cosine
feta	0.60	tomatoes	0.63
olive oil	0.64	capers	0.67
currants	0.65	hazelnuts	0.69
zucchini	0.68	sesame seeds	0.69
basil	0.69	chili peppers	0.70

Table 2: Most similar to *"cucumbers"*.

Word (GloVe)	cosine	**Word (s-GloVe)**	cosine
scallions	0.53	basil	0.60
radishes	0.53	peas	0.64
red onions	0.58	green beans	0.68
lettuce	0.61	beef broth	0.68
white vinegar	0.64	balsamic vinegar	0.71

These results show that the GloVe algorithm is capturing, as expected, the similarity between words that occur in similar contexts. Note that this is not the same as saying that they occur in the same recipes: lentils and brown rice may not occur together often, but when they occur separately they do so in the company of the same sorts of ingredients.

The s-Glove algorithm, however, is placing ingredients near to others that *are surprising when combined with the same sorts of ingredients*. GloVe suggests cucumbers are similar to radishes and red onions because (at least in our database) they occur in simmilar contexts, such as a variety of salads and Mediterranean dishes. S-Glove, however, finds cucumbers to be similar to ingredients like basil and peas, because it finds pairings like (cucumber, cocoa powder) and (cucumber, vanilla) to be highly similar to pairings like (peas, cocoa powder) and (peas, vanilla).

Table 3: Most similar to *"cayenne powder"*.

Word (GloVe)	cosine	**Word (s-GloVe)**	cosine
cumin	0.47	lemon juice	0.69
jalapenos	0.47	lemons	0.70
paprika	0.49	lime juice	0.74
chili powder	0.52	salt	0.74
garlic	0.55	celery	0.75

Table 4: Most similar to *"lentils"*.

Word (GloVe)	cosine	**Word (s-GloVe)**	cosine
brown rice	0.47	barbecue sauce	0.69
eggplant	0.47	steak	0.70
peas	0.49	brisket	0.74
kidney beans	0.52	ghee	0.74
barley	0.55	whiskey	0.75

While this is only a cursory validation, we can conclude from this that the s-GloVe algorithm is able to measure the similarity between when ingredients are found surprising. We hypothesise, and in the following section explore, that this property can be used to generate interesting suggestions for guiding users towards more novel content.

Results: surprise-surprise similarity

We used the s-GloVe vector model described in the previous section and calculated the pairwise vector subtraction between all pairs of ingredients. This represented every combination of two ingredients, even those that had not occurred in any of our recipes, as a 64-dimensional vector. In this we are inspired by the linear local substructures observed in other word embedding experiments (Agres et al. 2015; McGregor, Purver, and Wiggins 2016).

This space satisfies our notion of a surprise space defined earlier. It is a space of combinations of concepts, each with a location and a surprise rating, in which proximity implies similarity between *why those combinations are surprising*. To give an example, the closest concept to the surprising combination of mozzarella and brown sugar (excluding those that share either) is sausage and molasses. Despite their similar locations the two combinations have quite different surprise values: mozzarella and brown sugar is quite surprising (surprise $\approx$5), while sausage and molasses is only slightly surprising (surprise $\approx$2).

As an initial exploration of the potential of this space, we

Table 5: Most similar to *"apples"*.

Word (GloVe)	cosine	**Word (s-GloVe)**	cosine
raisins	0.46	ginger	0.68
cinnamon	0.53	icing sugar	0.71
nutmeg	0.54	walnuts	0.71
cranberries	0.55	currants	0.72
apple juice	0.57	cream	0.75

have implemented a simple – even trivial – surprise walk algorithm. Our motivation with surprise walks is to generate a sequence of combinations that can be incorporated into artefacts shown to a user. This sequence is intended to (gradually, perhaps with repeated exposure to artefacts containing each combination) guide the user towards being able to appreciate the "target" combination at the end of the sequence. That "target" is assumed to be outside of the "Wundt window" (i.e. off the right shoulder of the Wundt curve) for that user. It, along with a model of the user's familiarity with concepts in the domain, is the input to our model of surprise.

In our prototype we adopt a trivial synthetic user model: our prototype user is familiar with all surprises of less than 4 wows, as calculated by the method in (Grace and Maher 2016). This is based on the same co-occurrence matrix that is the input to the GloVe and s-Glove algorithms. Examples of combinations near this threshold are baking soda and tomatoes, apples and cumin, and lemongrass and walnuts. This threshold was chosen as it represents unusual but not (to the authors, at least) unheard of combinations, making it a good placeholder for the knowledge of a competent cook.

Our "surprise walk" algorithm, given a target surprise, first generates a list of the 25 nearest combinations. Those which are more surprising than the target are discarded. The system then iteratively greedily selects from that list the ingredient combination that most greatly reduces the surprise of the target without reducing it by more than a pre-defined "maximum surprise difference. In our experiments we set this threshold to 3 wows. If the selected combination is not familiar to the user then it becomes the new target and the greedy selection repeats. That means that if a target surprise is rated at 9 wows, then the system will pick the least surprising combination from the list of nearby combinations that is at least 6 wows, then repeat the process with a threshold of 3 less than that. At this point the combination would likely be less than our 4-wow threshold for the dummy user, and the sequence generation process would terminate.

This search is both greedy and naive. It is undirected, and would likely not work well with a more complex user model. A more nuanced approach would be to use a heuristic search algorithm like A* to find a path between the target and the user's "familiarity boundary". Despite its simplicity, this approach lets us explore the potential of traversing surprise spaces to generate goals for co-creative systems. Goal (re-)formulation has been suggested as a critical capacity for creative systems related to both autonomy (Jennings 2010) and metacreativity (Linkola et al. 2017).

Table 6 shows the result of a simple surprise walk on an ingredient combination that is only moderately surprising: bananas and basil. Both GloVe and s-Glove suggest one single combination as a sufficient stepping stone for the target combination. The suggested combination is familiar to our user (recall that our dummy user profile is familiar with all combinations of less than 4 wows) in both cases. This familiarity means that a co-creative system would likely only need to prompt the user with a few recipes before they are sufficiently primed as to appreciate bananas and basil.

GloVe suggested prompting the user with a combination of strawberries and thyme, highly literally similar to the target combination, but less surprising. Recipes involving this combination are typically pastries, jams[3], or cocktails. S-Glove suggested the less immediately obviously connected combination of applesauce and marjoram. Recipes involving this combination typically also involve pork, sausages, or game such as deer or partridge. s-GloVe considers applesauce and bananas to be quite similar (in terms of what they are surprising with), while GloVe does not. From this example it's hard to judge the quality of the two methods, although the difference in their approaches is clear.

Table 6: Surprise walks for *bananas and basil*.

Using GloVe:			
Ingredient 1	**Ingredient 2**	cosine	surprise
strawberries	thyme	0.34	2.51
bananas	*basil*	–	*4.69*
Using s-GloVe:			
Ingredient 1	**Ingredient 2**	cosine	surprise
applesauce	marjoram	0.42	3.01
bananas	*basil*	–	*4.69*

A similar case seems to be occurring in Table 7, which shows the recommended steps for a user to appreciate the highly surprising combination of parmesan and vanilla. This combination is found in a few unusual salads and cakes as well as one weird pasta recipe. GloVe suggests the user approach it by first trying artichokes and icing sugar, then capsicum and icing sugar[4]. As in the first example these ingredients occur in the same contexts as those in the target.

Table 7: Surprise walks for *parmesan and vanilla*.

Using GloVe:			
Ingredient 1	**Ingredient 2**	cosine	surprise
artichokes	icing sugar	0.29	2.23
capsicum	icing sugar	0.26	4.36
parmesan	*vanilla*	–	*7.14*
Using s-GloVe:			
Ingredient 1	**Ingredient 2**	cosine	surprise
mozzarella	figs	0.43	2.11
cheese	chocolate	0.39	4.25
parmesan	*vanilla*	–	*7.14*

By contrast s-Glove suggests that the user first try mozzarella and figs, then cheese and chocolate, then the target of parmesan and vanilla. Note that "cheese" here seems, on manual inspection of the dataset, to refer to the mild cheddar that is the typical "default" cheese in the Anglosphere. The left-hand side of this sequence seems to be based in literal similarity – all three are types of cheese, and two are prominent in Italian cuisine. This may be because all three are surprising in similar contexts (in addition to being literally similar), but it may also be an effect of the non-zero

[3]"Jam" most closely translates to "jelly" or "preserves" in North American English.

[4]"Capsicum" and "icing sugar" are "bell peppers" and "confectioner's sugar" in North American English.

weighting of unsurprising co-occurrences (as controlled by α in Equation 1). The right-hand side is more interesting, and begins to demonstrate the value of s-Glove. Chocolate is similar in context to vanilla, but not as similar as some of the other baking additives. Figs in turn are similar to chocolate, but not as similar as many other confections. What s-Glove provides is that the *combination* of cheddar and chocolate is supposedly like parmesan and vanilla *in terms of why it is surprising*. In other words, s-Glove suggests that someone could be better prepared for the high-surprise combination of parmesan and vanilla following this sequence than by following the literal recommendations of GloVe. We can as of yet offer no validation of this beyond our own opinions. Starting with mozzarella and figs (a common cheese-and-sweet-item combination found often alongside prosciutto, honey, or pistachios) and then proceeding to (cheddar) cheese and chocolate (less common, but still found in baked goods and more adventurous desserts) as a primer for trying parmesan and vanilla seems both plausible and palatable.

Table 8: Surprise walks for *worcestershire sauce and vanilla*.

Using GloVe:			
Ingredient 1	**Ingredient 2**	cosine	surprise
ketchup	icing sugar	0.25	3.45
paprika	icing sugar	0.25	6.17
worcestershire	*vanilla*	–	*7.82*
Using s-GloVe:			
Ingredient 1	**Ingredient 2**	cosine	surprise
kidney beans	chocolate	0.44	1.19
mustard	ice cream	0.44	4.87
worcestershire	*vanilla*	–	*7.82*

Table 9: Surprise walks for *soy sauce and chocolate*.

Using GloVe:			
Ingredient 1	**Ingredient 2**	cosine	surprise
ginger	cocoa powder	0.28	3.16
cabbage	cocoa powder	0.28	5.6
soy sauce	*chocolate*	–	*8.55*
Using s-GloVe:			
Ingredient 1	**Ingredient 2**	cosine	surprise
capsicum	jam	0.48	4.87
mushrooms	almond extract	0.43	6.22
soy sauce	*chocolate*	–	*8.55*

Tables 8 and 9 show similar trajectories to the first two examples. Both are combinations of sweet and savoury ingredients, a common theme among highly surprising combinations in our dataset. In Table 8 GloVe again goes with icing sugar as the closest ingredient to vanilla, this time pairing it with ketchup (as in some salad dressings) and then paprika (as in some moderately unusual baked goods). GloVe identifies similarly literal pairings in the soy sauce & chocolate case. S-GloVe, in Table 8 again suggests a sequence of seemingly unconnected but on deeper-inspection flavour-appropriate pairings: mustard ice-cream seems like excellent preparation for whatever unusual recipe could feature worcestershire sauce (a complex and pungent fermented condiment) and vanilla. Beans and chocolate are common combinations in Mexican and Mexican-inspired cuisine, but are still conceptually similar enough to mustard and ice-cream to serve as preparation.

In the final example s-GloVe appears to have selected what is (to the authors) a more unusual and less palatable combination, presented here for the purposes of showing that our preliminary models are far from flawless. Mushrooms are gastronomically quite similar to soy sauce, but the sequence of starting with capsicum and jam, then moving on to mushrooms and almond extract does not, to us, seem as appropriate a preparation for the combination of soy sauce and chocolate. Further developments in the construction of the surprise spaces, the representation of the data, and the algorithm for generating "surprise walks" are needed.

Discussion

In this paper we have presented a proof of concept for how a co-creative system might take planned, sequential action to change human opinion. To our knowledge, this is the first such work, with prior co-creative systems focussing on turn-taking and not conceiving explicitly of longer-term goals. The majority of current interactive creative systems typically do not engage in creative dialogues: they present, re-generate, and present again independently.

The capacity for planned, sequential interactions with creative systems raises a number of possibilities. Systems designed to educate less-expert users could introduce creative artefacts in sequences designed to broaden user horizons. Diverting creators away from low-novelty clusters of artefacts could also be useful outside explicitly educational contexts, given the prevalence of fixation in human designers (Jansson and Smith 1991). Similar approaches have been suggested in data mining contexts as a way to introduce users to the complex nuances of a dataset in an optimal way (Wagstaff et al. 2013). Alternatively, systems designed to diversify the behaviour of their users over time could have benefits for health and nutrition (Grace et al. 2017), using curiosity to overcome orthorexia and neophobia.

Any attempt to influence human behaviour with technology must necessarily be accompanied by an ethical framework. Investigations of what that might entail have arisen from the field of persuasive technology (Berdichevsky and Neuenschwander 1999; Verbeek 2006). Is it right to design systems that seek to change the desires of their users by manipulating their attention and curating their experiences? We, as creativity researchers, can decide that novelty and diversity are worthy of pursuit, but in doing so we implicitly devalue the traditional and the conservative. Luckily, in the contexts we see as near-future applications (education, design and nutrition, for example), it is simple enough to secure user consent in advance. In other contexts, such as using curiosity modelling to customise the news a user consumes, ethical minefields abound.

The most critical next step in this area of research will be to establish how "surprise walks" can be evaluated. The proof-of-concept results in this paper show that the concept has promise, but any further development will require more robust methodologies. One approach would be to devise laboratory experiments in which users are exposed to personalised sequences of artefacts and rate them for novelty, interest, value, etc. This would require "bootstrapping" a user model of knowledge and behaviour in a lab environment. Another approach would be to develop ways to quantify the difference between s-GloVe's "surprise space" and traditional conceptual spaces like GloVe. A final option, and one which remains a long-term goal of our research, would be to develop and evaluate an interactive system for diversifying behaviour by inspiring curiosity through surprise walks.

References

Agres, K.; McGregor, S.; Purver, M.; and Wiggins, G. A. 2015. Conceptualizing creativity: From distributional semantics to conceptual spaces. In *ICCC*, 118–125.

Berdichevsky, D., and Neuenschwander, E. 1999. Toward an ethics of persuasive technology. *Communications of the ACM* 42(5):51–58.

Berlyne, D. E. 1966. Curiosity and exploration. *Science* 153(3731):25–33.

Boden, M. A. 1996. *Dimensions of creativity*. Mit Press.

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.

Davis, N.; Hsiao, C.-P.; Singh, K. Y.; and Magerko, B. 2016. Co-creative drawing agent with object recognition. In *Twelfth Artificial Intelligence and Interactive Digital Entertainment Conference*.

Davis, N.; Karimi, P.; Maher, M. L.; and Grace, K. 2018. Creative sketching apprentice: Supporting conceptual shifts in sketch ideation. In *Design Computing and Cognition*.

Firth, J. R. 1957. A synopsis of linguistic theory, 1930-1955. *Studies in linguistic analysis*.

Gärdenfors, P. 2004. *Conceptual spaces: The geometry of thought*. MIT press.

Grace, K., and Maher, M. L. 2014. What to expect when youre expecting: the role of unexpectedness in computationally evaluating creativity. In *Proceedings of the 4th International Conference on Computational Creativity, to appear*.

Grace, K., and Maher, M. L. 2015. Specific curiosity as a cause and consequence of transformational creativity. In *ICCC*, 260–267.

Grace, K., and Maher, M. L. 2016. Surprise-triggered reformulation of design goals. In *AAAI*, 3726–3732.

Grace, K.; Maher, M. L.; Wilson, D.; and Najjar, N. 2017. Personalised specific curiosity for computational design systems. In *Design Computing and Cognition'16*. Springer. 593–610.

Gravina, D.; Liapis, A.; and Yannakakis, G. N. 2016. Constrained surprise search for content generation. In *Computational Intelligence and Games (CIG), 2016 IEEE Conference on*, 1–8. IEEE.

Harris, Z. S. 1954. Distributional structure. *Word* 10(2-3):146–162.

Jansson, D. G., and Smith, S. M. 1991. Design fixation. *Design studies* 12(1):3–11.

Jennings, K. E. 2010. Developing creativity: Artificial barriers in artificial intelligence. *Minds and Machines* 20(4):489–501.

Kiddon, C.; Zettlemoyer, L.; and Choi, Y. 2016. Globally coherent text generation with neural checklist models. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 329–339.

Liapis, A.; Yannakakis, G. N.; and Togelius, J. 2013. Sentient sketchbook: Computer-aided game level authoring. In *FDG*, 213–220.

Linkola, S.; Kantosalo, A.; Männistö, T.; and Toivonen, H. 2017. Aspects of self-awareness: An anatomy of metacreative systems. In *Eighth International Conference on Computational Creativity, ICCC, Atlanta*.

Macedo, L., and Cardoso, A. 2001. Modeling forms of surprise in an artificial agent. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 23.

McGregor, S.; Purver, M.; and Wiggins, G. 2016. Words, concepts, and the geometry of analogy. *arXiv preprint arXiv:1608.01403*.

Merrick, K. E., and Maher, M. L. 2009. *Motivated reinforcement learning: curious characters for multiuser games*. Springer Science & Business Media.

Newell, A.; Shaw, J. C.; and Simon, H. A. 1959. *The processes of creative thinking*. Rand Corporation Santa Monica, CA.

Pennington, J.; Socher, R.; and Manning, C. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 1532–1543.

Roemmele, M., and Gordon, A. S. 2015. Creative help: a story writing assistant. In *International Conference on Interactive Digital Storytelling*, 81–92. Springer.

Saunders, R., and Gero, J. S. 2001. Artificial creativity: A synthetic approach to the study of creative behaviour. *Computational and Cognitive Models of Creative Design V, Key Centre of Design Computing and Cognition, University of Sydney, Sydney* 113–139.

Schmidhuber, J. 2010. Formal theory of creativity, fun, and intrinsic motivation (1990–2010). *IEEE Transactions on Autonomous Mental Development* 2(3):230–247.

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):16.

Verbeek, P.-P. 2006. Persuasive technology and moral responsibility toward an ethical framework for persuasive technologies. *Persuasive* 6:1–15.

Wagstaff, K. L.; Lanza, N. L.; Thompson, D. R.; Dietterich, T. G.; and Gilmore, M. S. 2013. Guiding scientific discovery with explanations using demud. In *AAAI*.

Under the Super-Suit
What Superheroes Can Reveal About Inherited Properties in Conceptual Blending

Maria M. Hedblom and **Rafael Peñaloza** and **Giancarlo Guizzardi** and **Oliver Kutz**
KRDB, Department of Computer Science, Free University of Bozen-Bolzano, 39100, Italy
{firstname.lastname}@unibz.it

Abstract

Conceptual blending has been proposed as the cognitive machinery for concept generation. While computational approaches to conceptual blending have been implemented with some success, the automatic approaches still struggle to consistently produce concepts and blends that 'make sense' and have value. Mechanisms and optimality principles for blending have been introduced, yet their formal integration remains sparse. In this paper, we suggest to partly bypass this problem by identifying some *conceptual heuristics* for blending. This is done through a top-down analysis of three prototypical superheroes, an exemplary domain for conceptual blends and human imagination. We formalise the superheroes and backtrace their properties into their respective input spaces and from there map the inherited properties to cognitive theories for conceptualisation. It is our belief that computational blending systems could greatly benefit from conceptual heuristics for blending, identified in this top-down fashion. As a proof of concept of the identified superhero-blending heuristics, we blend the superhero 'Flowerman'.

Introduction

The nature of human creativity remains a hot topic of debate, and for research in artificial intelligence, it remains one of the most complicated of human phenomena to simulate. One theory that aims to explain the creative process is the theory of *conceptual blending* (Fauconnier and Turner, 1998). Building from a view of 'combinatorial creativity' (Boden, 1998) it proposes that it is by merging different mental spaces that novel concepts emerge. While there are other forms of creativity, this form has been given particular interest in the artificial intelligence community as it provides a concrete starting point to approach the complex research field of creativity (e.g., Kutz et al. (2014); Pereira and Cardoso (2006)).

One area in which conceptual blending is particularly perceptible is in comic books and the generation of *superheroes*. Comic books capture a range of human imagination and demonstrate conceptual blending as characters, settings and plots are heavily influenced by combinations of different conceptual domains. For instance, superheroes are often conceptual blends between humans and animals (e.g., Spiderman, Catwoman, and Antman) or humans and non-animated domains (e.g., Elastigirl, Aquaman, and The Human Torch). While there are many different kinds of superheroes, there seems to be a intuitive understanding (amongst humans) on which combinations of human and non-human attributes will "work;" i.e., be satisfactory in the context of superheroes, and which will not. For instance, there is no guarantee that, in the somewhat unlikely case that you are bitten by a radioactive spider, you remain a humanoid practically indistinguishable from your original form, only now enhanced with abilities such as 'wall climbing' and 'spider web shooting'. Without any 'blending control' it is equally likely that such a *Spiderman*-blend would encompass a creature with eight legs, a generously endowed bottom and with a taste for flies. An acquired-taste superhero that may not appeal to the average comic book reader (judge for yourself in Figure 1). This is a pivotal problem since in computational conceptual blending the number of possible blends grows exponentially in relation to the size, or detail, of the input spaces, in the context of which most of these blends will make little to no sense. The underlying rules for this intuitive understanding of what "works" have been introduced by Fauconnier and Turner (1998) as *optimality principles*. These are five[1] mental mechanisms that, when a person is "running the blend", automatically tweak the outcome to the most suitable blend for that context. While the work on formalising the rules behind these principles has been initiated

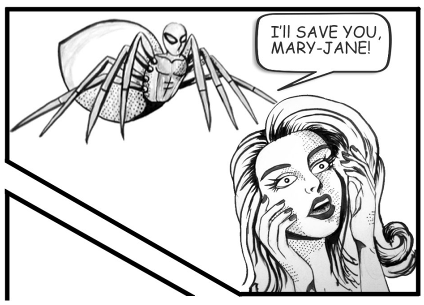


Figure 1: An alternative 'Spiderman' universe.

[1] Later, Fauconnier and Turner (2002) introduced three additional principles.

(Pereira and Cardoso, 2003b), they have also been deemed to be computationally difficult to capture as they are principles for certain structural patterns rather than concrete processes (Goguen and Harrell, 2010). For humans, these rules are more or less automatic. However, for computational conceptual blending, they are still a bottleneck that requires serious attention.

This paper aims to bypass this problem by identifying, in a top-down fashion, some of the blending heuristics for superheroes that tell us something about the essence of 'super', what it is that those heroes have in common. By formally exploring a few prototypical superheroes, we assess their most prominent features, backtracing them from the blended spaces into the conceptual spaces from which they were merged (or emerged). We believe that identifying such, sometimes domain-dependent, inherent mechanisms will provide useful information to increase the performance of state-of-the-art computational approaches to conceptual blending.

The paper is structured as follows: First, conceptual blending and a number of theories that help to uncover the underlying mechanisms behind conceptualisation are introduced. Second, a few well-established superheroes are dissected into their input spaces and inherited properties from each input space, followed by identifying and introducing the derived conceptual heuristics for the blending process. Third, a 'proof of concept' superhero is generated using the heuristics. The paper ends with a discussion and related work as well as speculations on the potential impact of, and interconnections between, uncovering the underlying mechanisms in computational blending vis a vis an analysis of these mechanisms in the light of formal ontologies and design patterns.

Setting the Scene

Conceptual Blending and 'Running the Blend'

Inspired by the principles of analogical reasoning, in which one domain carries information over to another less information-rich domain, Fauconnier and Turner (1998) introduced conceptual blending. The gist of the framework is that information stored in conceptual spaces are blended into a novel blended space through selective projection. During blending certain emergent features in the blend may appear without direct transfer from either input space, but rather develop as a consequence of the blended spaces particular properties. This emergence is the result of the mechanisms behind 'running the blend'.

While the mechanisms that underlay these principles for emergence are largely unknown, the principles have been specified to some extent. For our current purposes we limit ourselves to report on three such mechanisms: First, *composition* ensures, for instance, that certain part-whole relationships are maintained in the blend regardless of what information was transferred, e.g., in the case of animal blending that a head is attached to a neck or that a stomach is on the inside of the body. Second, *completion* is the principle of 'filling in the blanks', i.e. the blend might inherit insufficient information from the input spaces for the 'blend' to make sense and therefore emergent properties arise. Third, *elaboration* develops the blend through imaginative mental simulation given the already accepted rules and principles of the blend space. The emergence process might go on indefinitely with new completion structures, as well as with new laws and principles, emerging through the continuation of elaborative processes (Pereira and Cardoso, 2003b; Fauconnier and Turner, 1998).

In humans, these processes appear to be without much mental effort. Rich conceptual understanding provides excellent grounds for novel concepts to emerge, and contextual awareness ensures that the novel conceptual blends also are 'appropriate' and valuable, defining the blending process as a creative process as it is both novel and valuable (Runco and Jaeger, 2012). For computational creativity, the conceptual blending process has several issues in need of attention. One major problem is directly related to the richness of human conceptual knowledge and its intuitive understanding for appropriate combinations. While computer systems are ever increasing in data capacity, the progress in producing systems that consistently make sense is slow. In fact, as the amount of information in the input spaces grows richer, the number of generated blends grows exponentially. Any computational system dealing with conceptual blending needs to apply a series of rules and heuristics to avoid this. Some of the contributions to computational conceptual blending that outline paths towards addressing this issue include Hedblom, Kutz, and Neuhaus (2016); Eppe et al. (2018); Pereira and Cardoso (2003b).

Another important aspect to note is that it is tempting to assume that all (complex) concepts are the result of conceptual blending. While this might be true on a deeper level, it is not a fruitful assumption for most common day scenarios. Take the superhero 'Batman' from the DC Universe. Initially, one could argue that Batman is the blend of the input spaces Bats and Man. However, on closer inspection, Batman has inherited a rather limited number of properties from Bats. The only major influence is a visual analogy between his suit and a bat, and additionally a few wordplays like 'Batcave' and 'Batmobile'. On a conceptual level, Batman does not have any pertinent attributes associated with bats. Compare this to Marvel's Spiderman, a man who after being bitten by a radioactive spider is 'enhanced' with characteristics and abilities found in spiders.

So while it might be tempting to immediately assume conceptual blending rules all superheroes, there seems to be a significant quality distinction that needs to be addressed.

Identifying the Superhero through Cognitive Theories

Another difficult question involved with computational conceptual blending has little to do with the blending process itself, but rather with the structure of knowledge and conceptualisation. Human conceptual knowledge is vast and not only is it difficult to capture its span but it is uncertain how the mind structures it in the first place. Here, we present a few theories addressing how humans are thought to identify the meaning of things and which, we argue, are particularly relevant for conceptual blending of superheroes.

Conceptual Metaphor Theory: Related to conceptual blending is the research field on conceptual metaphors[2] (Lakoff and Johnson, 1980). Similar to conceptual blending, conceptual metaphor theory aims to undress analogous expressions to the conceptual core and transfer essential information from one domain to another. The theory rests on the basis that there exists a limited number of conceptual skeletons that humans use to structure their knowledge (Kövecses, 2010). A prototypical conceptual metaphor is: "DARK is BAD", which is a common method to depict the villains in comic books. For instance, how Spiderman's outfit turns black when he is infused by the supervillain Venom[3].

Image Schemas: One theory that aims to ground the conceptual metaphors into conceptual building blocks is the theory of image schemas (Johnson, 1987; Lakoff, 1987). Building from the idea of embodied cognition, the theory presupposes that there exists a limited number of spatio-temporal relationships learned from the sensorimotor processes in early infancy that are used to reason about events and the surroundings (Mandler, 2004). For instance, a table offers the image schema of SUPPORT and a house CONTAINMENT. This information then can be transferred to increasingly abstract scenarios through metaphors and associations. In the conceptual metaphor "UP is GOOD," VERTICALITY, or its dynamic version UP_DOWN, is the image schema at work. Image schemas are also suggested to be one of the core components in analogical reasoning and conceptual blending (Hedblom, Kutz, and Neuhaus, 2016). In comic books, both image schemas and conceptual metaphors are key components in encouraging particular interpretations such as who is good and who is evil as well as representing movement and sounds that are not possible in the static comic book format (Potsch and Williams, 2012).

Affordances: The hypothesis that image schemas construct the smallest conceptual building blocks is further supported by the theory of affordances[4] as image schemas have been suggested to be categorised as clusters of affordances (Galton, 2010). Affordance theory was introduced by Gibson (1977) and suggests that the meaning of objects, and concepts as a whole, can be described through the affordances that they offer to an agent. For example, a bed is a bed because you can 'sleep in it,' and a coffee cup is a coffee cup because you can 'drink coffee from it.' In relation to image schemas, the bed has the SUPPORT image schema and the cup has CONTAINMENT. This point of view provides a straightforward method to look at the essential properties of concepts. Within the affordance framework, a hero would be a hero because they offer the affordance of 'rescue' and a superhero would simply be a hero that offers rescue through some 'supernatural' means.

[2]Also called cognitive metaphors, or more specifically image schema metaphors (Kövecses, 2010).

[3]There are plenty of counterexamples for this. For instance, the villains in Batman are generally a rather colourful bunch, whereas Batman himself is rather grim.

[4]In this paper, we exclusively view affordances in the Gibsonian sense.

Recognition-by-parts: While affordances have lots to offer as a theory to the essential core of objects and concepts, there are naturally other characteristics that are of importance for the essence of objects. For instance, for all CONTAINMENT, there needs to be an inside, an outside and some sort of border. This naturally translates to a set of visual and physical characteristics. Recognition-by-parts was introduced as a means to break visual features into smaller geometric blocks called geons (Biederman, 1987). Hence, we can identify a cup because it is composed of the geons a 'hollow cylinder' and a 'handle'. Regarding superheroes and other roles, the visual features might not be as easy to core down to visual components as simple as geons, but there are visual cues of such importance that they are seen as part of the hero's essence. For instance, most would be able to identify a rough silhouette of a superhero based on the physical shape, the cape, the inside-out underwear and boots.

Prototype Theory: This leads to another important theory for the nature of things, namely prototype theory (Rosch, 1973). It suggests that for all categories (e.g., superheroes) there is a prototype to which more or less all members of that group should show some resemblance. A prototypical superhero like 'Superman' ensures that all members are similar to his properties. Superheroes that venture too far from the prototype do not qualify as members of that category.

Essentialism: From the point of view of essentialism, Neuhaus et al. (2014), while blending monsters, argue that one essential criterion is that the resulting blend needs to be 'scary.' For superheroes, a corresponding essential property is that of 'being (morally) good.'[5] This means that based on conceptual metaphors and stereotypes associated with goodness such as "GOOD is BEAUTIFUL", a superhero gains some (if not all) the conceptual information we attach to 'goodness.' This includes attributes such as beauty, generosity, wisdom, and a range of other 'generally positive' features that in reality might have little to do with goodness in itself. Arguably, it could be the case that the features associated with goodness in themselves do not need to be inherited, but rather that their conflicting attributes are unwelcome. For instance, a superhero (that follows the conventional 'goodness' model) *may not* be ugly, selfish, or stupid, rather than imposing that they *have to* be beautiful, generous and wise.[6] In fact, according to a historical analysis of the physical appearance of comic superheroes, attractiveness appears to not only be important but pivotal (Avery-Natale, 2013).

Following the presented theoretical framework for the conceptualisation of things and roles and in the light of conceptual blending we proceed to ask: *How is a superhero created?*

[5]There are unconventional cases in which superheroes are not, in the classic sense, intrinsically good, e.g., Hellboy and Deadpool. For now, we focus on the most prototypical superheroes, where this property holds.

[6]Naturally, there are counterexamples to this as well, where the attractiveness of the hero is somewhat questionable, e.g., Thing and Man-Bat, but often these are already somewhat 'dehumanized' by their names.

Heuristics for Blending Superheroes

Carving the 'Superhero Mould'

There are several attributes and requirements that guide the selection of the properties while performing blending. In the setting of this paper, the blend is by definition required to be a 'hero that is super,' hence the outcome is required to follow a hero template. One definition of a hero is "a person noted for courageous acts or nobility of character."[7] This means that any hero is an animated entity (i.e., a person) who needs to at a minimum have the attributes 'courage' and 'goodness'. As the blended space is intended to be a superhero, further distinctions are needed. A superhero can be defined as "a hero, especially in children's comic books and television cartoons, possessing extraordinary, often magical powers."[8] The relevant distinction between a hero and a superhero is the addition of 'extraordinary powers.' This distinction is of vital importance as it ensures that at least one for-humans unconventional power is inherited from the non-human input space. However, note that this is not necessarily a non-human ability such as flight, or x-ray vision; it can also take the expression of a human ability blown out of proportions, e.g., *The Flash*, who inherits 'superspeed' from the input space lightning or in the case of *Spiderman*, 'superstrength' as spiders are assumed to carry up to 20 times their own bodyweight. Note that this kind of treatment of already existing human powers is done through the image-schematic transformation of SCALING.

As we have argued that superheroes are blends we need to define the mould by which superheroes are blended. Based on the definitions above and the ideas behind essentialism we can infer that the superhero mould needs to have the following characteristics: 'Courage' and 'Goodness;' and the ability: 'at least one extraordinary power.'

When looking at prototype theory and recognition-by-parts, the visual attributes of a superhero appear equally important. Superheroes tend to be attractive, their outfits are typically made in tight spandex, have both capes and inside-out underwear and are generally colourful with symbols representing their 'core identity'. Spiderman has a spider, Superman has a big S and Batman has an outfit that is entirely bat-inspired. Therefore, the prototype hero also requires the visual attributes: 'attractive' and 'wears suit with emblem.'

Identifying the prototype superhero, or the superhero mould, is of great importance as it is used to evaluate and eliminate conflicting attributes in the blended superhero. This means that the blended superhero will most often (if not always) be forced to fit into the superhero mould. If it does not fit, it might not be considered a 'true' superhero. Based on this reasoning we define the minimum requirements for a prototype hero in the following, using DL syntax:

$$\begin{aligned}\mathsf{Superhero\_Mould} \equiv\ & \mathsf{Person} \sqcap \mathsf{Attractive} \sqcap \mathsf{Courageous} \sqcap \\ & \mathsf{Good} \sqcap \exists \mathsf{has.ExtraordnaryPower} \sqcap \\ & \exists \mathsf{wears.Suit} \sqcap \exists \mathsf{has.Emblem}\end{aligned}$$

[7] http://www.dictionary.com/browse/hero. Retrieved February 14, 2018

[8] http://www.dictionary.com/browse/superhero, Retrieved February 14, 2018

Conceptual Modelling of A Few Prototypical Superheroes

As we are verging on uncovering what lies underneath the superhero costume, our method for analysing the blending process is by backtracing from a few well-established superheroes, to identify the input spaces and the attributes and abilities that they have inherited from each space. We look closer at the Marvel Comics' heroes Spiderman, Black Panther and The Human Torch. Below, each superhero is formalised.[9]

Spiderman: Under Spiderman hides Peter Parker, an intelligent science student who after being bitten by a radioactive spider acquires several abilities associated with affordances that particular characteristics of spiders offer. Some of the most prominent ones are that he can climb walls, he shoots spider webs, and has increased senses that provide him with a 'spider sense' to perceive his surroundings.[10] In addition, his human strength and speed are through SCALING blown up to that of a spider in human size. Using description logics, we can formalise Spiderman as:[11]

$$\begin{aligned}& \mathsf{YoungMan} \sqcap \mathsf{Intelligent} \sqcap \mathsf{Good} \sqcap \mathsf{Courageous} \sqcap \\ & \exists \mathsf{climbs.Wall} \sqcap \exists \mathsf{expells.Web} \sqcap \mathsf{SuperStrong} \sqcap \\ & \mathsf{has.SpiderEmblem}\end{aligned}$$

The Black Panther: The Black Panther is T'Challa who, by a shamanistic connection to a Panther God, acquires several catlike characteristics. Some prominent ones are acute senses, enhanced strength, speed, agility, stamina, durability, healing, and reflexes. In addition he has the claws of a cat which affords him the ability of climbing VERTICALITY and using them as weapons in direct combat.

$$\begin{aligned}& \mathsf{Man} \sqcap \forall \mathsf{hasColour.Black} \sqcap \forall \mathsf{hasWeapon.Claws} \sqcap \\ & \mathsf{Good} \sqcap \mathsf{Courageous} \sqcap \mathsf{Agile} \sqcap \\ & \exists \mathsf{needs.Oxygen} \sqcap \forall \mathsf{eats.}(\mathsf{Meat} \sqcup \mathsf{Vegetable}).\end{aligned}$$

The Human Torch: Johnny Storm from the Fantastic Four is an example of a non-animal blended superhero. He gains his 'superpowers' from the inanimate input space Fire. The Human Torch is able to envelop his body in flames (i.e. CONTAINMENT) which also gives him the power to fly, motivated through the physics behind 'how flames rise' (the, from the 'just-human' point of view, 'supernatural' combination of the image schemas VERTICALITY and SOURCE_PATH_GOAL). Additionally, he can produce balls of fire. Simultaneously he reacts weak to the same things

[9] We acknowledge both the male-dominance and their limited formalisations but argue that we have captured some of the most relevant features that make each particular superhero unique.

[10] This might actually not be a spider skill in itself, however, it could be interpreted as the result of sensing the surroundings as a spider senses activity in their nets.

[11] Note that we also specify:

$$\begin{aligned}\mathsf{YoungMan} &\equiv \mathsf{Man} \sqcap \mathsf{hasAge.}(\leq 25) \\ \mathsf{Person} &\equiv \mathsf{Man} \sqcup \mathsf{Woman}\end{aligned}$$

that fire is, as water 'extinguishes' him and lack of oxygen hinders his powers.

$$\text{YoungMan} \sqcap \exists \text{needs.Oxygen} \sqcap \text{Good} \sqcap \text{Courageous} \sqcap \exists \text{hasCapacity.Empathy} \sqcap \exists \text{hasWeapon.FireBall} \sqcap \exists \text{diesFrom.}(\text{Suffocation} \sqcup \text{Freezing} \sqcup \text{Drowning})$$

The input spaces

Man: By the definition presented above, a hero was required to be a person. In the three examples above, all heroes were male so we require that input space to be described in more detail. A human male is an animated creature with a humanoid form, with two arms, two legs, a torso and a head, which walks upright (SOURCE_PATH_GOAL). It has high levels of intelligence and is capable of empathy (which we treat as a prerequisite for developing courage and goodness while running the blend based on the optimality principles). It requires oxygen to breathe and food to eat.

$$\text{Man} \equiv \text{Person} \sqcap \text{Male} \sqcap \exists \text{hasShape.Humanoid} \sqcap (=2.\text{hasLegs}) \sqcap \exists \text{hasCapacity.Empathy} \sqcap \exists \text{needs.Oxygen}.$$

Spider: A spider is an eight-legged arachnoid, capable of carrying 20 times its own weight and through its characteristics affords the ability of wall-climbing. Additionally, it is able to expel webs and it injects venom into its victims. To many humans, spiders are perceived as malicious animals potentially due to them being cannibalistic predators, their alien visual appearance or their physical threat to humans.

$$\text{Spider} \equiv \text{Arachnoid} \sqcap (=8.\text{hasLegs}) \sqcap \text{Malicious} \sqcap \exists \text{climbs.Wall} \sqcap \exists \text{expels.Web} \sqcap \forall \text{injects.Venom} \sqcap \text{Strong}$$

Panther: A black panther is a particular kind of felid, characterised by its black colouring, speed, grace, and strength. It is a carnivorous quadruped, which hunts larger prays for survival. They are capable of jumping very high, and can maintain a high speed for a long period of time. They are dangerous to humans and can be considered fearsome.

$$\text{Panther} \equiv \text{Felid} \sqcap \forall \text{hasColour.Black} \sqcap (=4.\text{hasLegs}) \sqcap \text{Fearsome} \sqcap \exists \text{needs.Oxygen} \sqcap \forall \text{eats.Meat}$$

Fire: Fire is the result of combustion, releasing heat, light, and various chemical components. It is enabled by the presence of oxygen in the environment, and can die through suffocation, freezing, or drowning. It is a chemical reaction, which can burn, but also stimulate growth.

$$\text{Fire} \equiv \text{ChemicalReaction} \sqcap \text{Hot} \sqcap \exists \text{needs.Oxygen} \sqcap \forall \text{rise.Flames} \sqcap \exists \text{diesFrom.}(\text{Suffocation} \sqcup \text{Freezing} \sqcup \text{Drowning})$$

This leads us to uncover the inherited properties that are unique to the individual superhero. In the next section, we look closer at what this means.

Inherited Properties

By separating the input spaces from the blended superheroes we can identify the nature of which properties are inherited from each input space.

What can be determined is that all three superheroes inherit the personality characteristics from the human input spaces. They all remain the same people with their intelligence and their morals intact, but they are enhanced by being provided with increased strengths and inhuman abilities. Spiderman inherits properties that affords him with abilities to be able to attach himself to walls, and cast webs to capture enemies in and to be able to move around in three-dimensions. Basically, the SOURCE_PATH_GOAL image schema found in ordinary human behaviour has been enhanced to include also a vertical dimension. Similarly, Black Panther is enhanced with the gracious strength and agility found in large cats from the cat input and is provided with claws. The optimality principles for blending ensures that the presence of such characteristics are also translated into affordances and abilities, meaning that Black Panther's preferred weapon is martial art with a bit of claw. Notable also is that the generic space here ensures that the 'black' identity of the superhero is preserved. The Human Torch has been awarded the ability of flight when he is enclosed in flames. This is inspired from the input space Fire based on the idea that flames rise (the VERTICALITY image schema). Moreover, interestingly he also inherits handicaps as a consequence of this blending process. While both humans and fires require oxygen to function, a fire cannot be lit under water, which is transferred to The Human Torch and is often used as a weapon against him. Regarding their visual appearance, it is obvious that the essence of being human is preserved based on the Superhero prototype requiring them to remain "people", however, their outward appearances are heavily influenced by the non-human input space. Spiderman's suit carries a spider emblem, Black Panther's suit is heavily cat inspired and The Human Torch wears a red and yellow suit resonating with the colours of burning flames.

Based on these observations we proceed to build conceptual heuristics concerned with how to create a superhero of our own making.

The Superhero Recipe

1. Choose Input space 1 (I_1): a 'human' conceptual domain and define characteristics e.g., female, male, age, ethnicity, etc.
2. Choose Input space 2 (I_2): a conceptual domain of interest; e.g., an animal, an element, etc.
3. Specify the superhero prototype and form the mould for the blended space. Identify and generalise:[12]
   (a) Visual features: e.g., wears colourful cape and suit, muscular etc.
   (b) Characteristics: e.g., good, patient/impulsive etc.

[12] Note that the superhero mould's characteristics are examples of 'slots to be filled' and not criteria. Any kind of superhero could be built that does not need to follow the prototypical goodness-model used in this article.

(c) Abilities: e.g., speed, strength, flight, ex-ray vision etc.

4. Cross-identify visual features, characteristics and abilities between I_1 and I_2. Generate the generic space based on this.
5. Identify personality traits and characteristics from I_1 and transfer it to the blend.
6. Identify abilities based on affordances and image schemas in I_2 and transfer those abilities to the blend.
7. Remove all attributes that are in conflict with the identified superhero prototype, e.g., 'evil' cannot be present if 'goodness' is part of the prototype.
8. Run the blend through the blending optimality principles to maximise the success of the blend.

Based on our general analysis and the workflow presented above, it is now possible to 'build' new superheroes following these heuristics.

Proof of Concept: Introducing 'Flowerman'

In the previous sections, the blending process of superheroes was backtraced to identify some underlying blending heuristics guided by a number of theories on concept formation and meaning generation. In this section we introduce *Flowerman*, a proof-of-concept hero based on these heuristics.

Step 1: we choose to build an adult 'male' superhero, hence Input space 1: Man.

$$\mathsf{Person} \sqcap \mathsf{Male} \sqcap \exists \mathsf{hasShape.Humanoid} \sqcap (=2.\mathsf{hasLegs}) \sqcap \exists \mathsf{hasCapacity.Empathy} \sqcap \exists \mathsf{needs.Oxygen} \sqcap \exists \mathsf{eats.Food}$$

Step 2: we choose the complementary conceptual domain, input space 2, based on Flower.

$$\mathsf{Plant} \sqcap \mathsf{Beautiful} \sqcap \mathsf{MorallyNeutral} \sqcap \exists \mathsf{has.Petals} \sqcap \exists \mathsf{hasCapacity.ejectSeeds} \sqcap \mathsf{Grows} \sqcap \exists \mathsf{needs.CarbonDioxide} \sqcap \exists \mathsf{eats.Sunlight}$$

Step 3: we identify the prototypical goodness-model superhero as defined in the superhero mould above. This means the superhero must wear a suit with emblem, be attractive, be good and courageous as well as have an extraordinary power.

Step 4: by mapping and generalising the structure in Man and Flower the following generic structure appears. The generic space is as follows:

$$\exists \mathsf{hasCapacity.Y} \sqcap \exists \mathsf{needs.Z} \sqcap \exists \mathsf{eats.X}$$

Steps 5 and 6: From the Man we preserve the human attributes, and from Flower abilities based on affordances are preserved so that together they construct the blended space. The blended space is thus *Flowerman*:[13]

$$\mathsf{Person} \sqcap \mathsf{Attractive} \sqcap \mathsf{hasCapacity.Empathy} \sqcap \exists \mathsf{hasCapacity.ejectSeeds} \sqcap \exists \mathsf{wears.petalsSuit} \sqcap \exists \mathsf{has.FlowerEmblem} \sqcap \exists \mathsf{eats.}(\mathsf{Sunlight} \sqcup \mathsf{Food}) \sqcap \exists \mathsf{needs.}(\mathsf{Oxygen} \sqcup \mathsf{CarbonDioxide})$$

[13]Note that beautiful and attractive are treated as synonyms.

Step 7-8: The blended concept *Flowerman* is matched to the prototype Superhero in order to inherit the human form and the 'hero' attributes such as goodness and courage from the input space Man which is acquired when running the blend based on the capacity for empathy. From the Flower he inherits the abilities to eject seeds, which turns into a "Seed-Gun" of some sort through elaboration. The suit from the superhero mould is merged with the 'petal-dress' of the flower to generate a 'suit of petals.' Additionally, *Flowerman* has the ability to 'eat' sunlight, potentially through chlorophyll present in green skin, a feature that would be developed as an emergent property through *composition* and *elaboration* and he can choose to breathe either oxygen or carbon dioxide.

Whether *Flowerman* will be the next big thing in the comic book world is up for time to tell. However, the procedure by which he was made could help to advance the computational conceptual blending scene. Here we have taken potential aspects of blending superheroes into account and manually used the identified heuristics to create a novel superhero. If a computer system that handles logical rules such anti-unification as seen in the analogy engine and conceptual blender Heuristic-Driven Theory-Projection (HDTP) Schmidt et al. (2014); Guhe et al. (2011) and Structure Mapping Engine (Forbus, Falkenhainer, and Gentner, 1989), or the computational conceptual blender Divago (Pereira and Cardoso, 2006) provided with a similar script the blending outcome may be shown to be improved.

Discussion and Related Work

Comic books have been shown to be a good playground for identifying conceptual blends. In comparison to looking at individual superheroes, as done in this article, Szawerna (2012) makes an in-depth analysis of the complete blended universe in the comic book *Watchmen* by studying cross-domain parallels between the real US politics and foreign affairs in the fictive world with superheroes. Similarly, Forceville (2016) presents the role of conceptual blending in cartoons and comic strips to illustrate how meanings not directly present in the comic strips are transferred through conceptual metaphors and conceptual blending mechanisms. His work also strengthens the hypothesis this paper identified, namely, that the role of affordances and image schemas plays a central role when inheriting valuable information from the non-human input space. This is also the conclusion found by Potsch and Williams (2012) who points out how image schemas are directly related to how conceptual information regarding movement is depicted in the still frames of the comic format and the work on computational conceptual blending by Hedblom, Kutz, and Neuhaus (2016).

Another bottom-up approach to analysing the blending process is the work by Neuhaus et al. (2014). By looking at formal conceptual blending they investigate the automatic generation of monsters by merging OWL formalisations of animals together. Their work rests on the foundation that the blended monster needs to satisfy the criterium of being 'scary'. This relates to the initial criteria of superheroes having 'courage' and having 'extraordinary' abilities of some sort. Similarly, the work by Pereira and Cardoso (2003a) demonstrates how the computational blender Divago can blend the concept of 'horse' with 'bird' to generate a pegasus. The Divago system is particularly interesting as it has initiated the work on formalising the optimality principles.

These studies differ from this article by either simply analysing the state of blending in comics, or by approaching the blending processes in a bottom-up fashion. Our attempts to identify some blending heuristics for superheroes took the opposite direction, by first analysing the superheroes top-down to identify some criteria and based on this generate a new superhero bottom-up. While the approach does show promise in identifying some core heuristics for conceptual blending that could be used in computational approaches, the work here suffers from two major disadvantages. First, as the formalisation for both the input spaces and the superhero blends are handcrafted, they are subject to errors and favourable interpretations that might not be present in a more natural scenario. Second, the superhero blending is based on the notion of a prototypical superhero based on the goodness-model. As has been discussed, there are several superheroes that venture out from the norm, with questionable morals, visual appearance that verges on being inhuman, and characteristics that do not fit the here identified superhero mould. That said and within that prototypical domain, an interesting find is that blended superheroes often gain the abilities, based on affordances and image schemas, from the non-human input and the characteristics from the human input. The inherited visual appearance is something that is partly based on the superhero prototype, namely that they have to be attractive with strong humanoid bodies while the non-human input space offers less intrusive characteristics to be inherited, such as colour schemes for the Super-Suit or icons and symbols that are associated with that particular superhero, e.g., Spiderman's spider logo on his suit, or Black Panther's catlike suit.

As argued in Neuhaus et al. (2014), the steering of the automatic construction of blends requires a mix of requirements: (ontological) constraints/consistency requirements and consequence requirements. These are heavily domain-specific, and we have here presented the core of a requirement theory for the automation of the superhero mould.

The Road Ahead: Conceptual Blending from an Ontological Perspective

In this paper, we assume that the concepts (representing monadic types or unary predicates) that participate in blending operations all stand in the same ontological footing. However, as discussed in Guizzardi (2005), from an ontological perspective, different categories of concepts classify entities in completely different manners. For instance, if we take a particular individual named *Peter Parker*, he can be (at the same time or across time) classified under the concepts Person, Adult Man, Reporter, and Physical Object, among others. However, it is not the case that all these concepts classify *Peter Parker* in the same manner. First of all, Person is a **Kind** (or **Substance Sortal**) and, as such, it captures the essential properties of the entities it collects and provides principles of individuation, cross-world identity and persistence for them (see Guizzardi (2005)). In contrast, Physical Object is an example of a **Non-Sortal** concept and, as such, one which cannot provide a uniform principle of identity for its instances and, hence, which represents properties that occur in individuals of multiple **Kinds**. Furthermore, concepts like Adult Man, Student or Reporter represent **Anti-Rigid Sortals**, i.e., concepts that represent contingent properties of entities of a particular **Kind** (in this case, Person). Nonetheless, still under this category, we have concepts that capture *intrinsic and contingent properties* of entities of a given **Kind** (e.g., being an Adult Person is being a Person who has the intrinsic contingent property of being in a certain developmental phase). These are called **Phases**. On the other hand, we have concepts that capture *contingent but relational properties* of entities of a given **Kind** (e.g., being a Reporter is being a Person who has the contingent and relational property of working for a news organization). These concepts are called **Roles**.

Now, the conceptual blending operations discussed in this article seem to follow a particular ontological recipe: (1) select two **Kinds** (e.g., Person and Spider); (2) one of these **Kinds** will be preserved as the **Kind** of the resulting concept (e.g., Person) and the other one will be used to abstract a **Non-Sortal** concept capturing the characteristics that are necessary for the intended blending (e.g., Arachnoid-Entity). Notice that Arachnoid-Entity is indeed a **Non-Sortal** as it classifies entities of multiple **Kinds** (i.e., entities of the **Kind** Person and of the **Kind** Spider). Moreover, it is an example of a semi-rigid **Non-Sortal** (i.e., a so-called **Mixin**, see Guizzardi (2005)), as it defines properties that are essential for some of its instances (i.e., for Spider, which are necessarily Arachnoid-Entities), while being contingent for other instances (i.e., instances of people are only contingently Arachnoid-Entities). In other words, for example, Peter Parker existed without having those properties and can still survive maintaining its identity (i.e., exist as the same *Person*) in case he loses these properties; (3) create a concept that specialises by intersecting the **Kind** selected in (1) with the **Mixin** produced by abstraction in (2). The result will typically be an **Anti-Rigid Sortal** (e.g., a **Phase**, if we think of the concept Man-with-Spider-Powers, or **Role**, if we think of Spider-Man, i.e., as Man-with-Spider-Powers who acts as a hero, bearing certain responsibilities w.r.t. a community, etc.).

For future work, we intend to systematically investigate the connection between the conceptual blending operations discussed here with the rich literature on categories of concepts/types as proposed in the area of formal ontology. This will establish a connection between theories of blending and those of *Ontology Design Patterns* as discussed in, for example, Kutz et al. (2016); Ruy et al. (2017). For doing that, we will also need to extend our formal characterisation of these operations, since the characterisation of these different categories of types necessarily requires the treatment of modal notions (e.g., rigidity or relational dependence).

References

Avery-Natale, E. 2013. An analysis of embodiment among six superheroes in dc comics. *Social Thought & Research: A Continuation of the Mid-American Review of Sociology* 71–106.

Biederman, I. 1987. Recognition-by-components: a theory of human image understanding. *Psychological review* 94(2):115.

Boden, M. A. 1998. Creativity and artificial intelligence. *Artificial Intelligence* 103:347–356.

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., and Turner, M. 2002. *The Way We Think: Conceptual Blending and the Mind's Hidden Complexities*. Basic Books.

Forbus, K.; Falkenhainer, B.; and Gentner, D. 1989. The structure-mapping engine. *Artificial Intelligence* 41:1–63.

Forceville, C. 2016. Theories of conceptual metaphor, blending, and other cognitivist perspectives on comics. In Cohn, N., ed., *The Visual Narrative Reader*. London: Bloomsbury.

Galton, A. 2010. The Formalities of Affordance. In *Proceedings of ECAI-2010 Workshop on Spatio-Temporal Dynamics*, 1–6.

Gibson, J. J. 1977. The theory of affordances, in perceiving, acting, and knowing. towards an ecological psychology. In Shaw, R., and Bransford, J., eds., *Perceiving, Acting, and Knowing: Toward an Ecological Psychology*. Hillsdale: NJ: Lawrence Erlbaum. 67–82.

Goguen, J. A., and Harrell, D. F. 2010. Style: A computational and conceptual blending-based approach. In *The structure of style*. Springer. 291–316.

Guhe, M.; Pease, A.; Smaill, A.; Martinez, M.; Schmidt, M.; Gust, H.; Kühnberger, K.-U.; and Krumnack, U. 2011. A computational account of conceptual blending in basic mathematics. *Cognitive Systems Research* 12(3-4):249–265.

Guizzardi, G. 2005. *Ontological Foundations for Structural Conceptual Models*. Universal Press.

Hedblom, M. M.; Kutz, O.; and Neuhaus, F. 2016. Image schemas in computational conceptual blending. *Cognitive Systems Research* 39:42–57.

Johnson, M. 1987. *The Body in the Mind: The Bodily Basis of Meaning, Imagination, and Reason*. Chicago and London: The University of Chicago Press.

Kövecses, Z. 2010. *Metaphor: A Practical Introduction*. Oxford University Press, USA.

Kutz, O.; Bateman, J.; Neuhaus, F.; Mossakowski, T.; and Bhatt, M. 2014. E pluribus unum: Formalisation, Use-Cases, and Computational Support for Conceptual Blending. In Besold, T. R.; Schorlemmer, M.; and Smaill, A., eds., *Computational Creativity Research: Towards Creative Machines*, Thinking Machines. Atlantis/Springer.

Kutz, O.; Neuhaus, F.; Hedblom, M. M.; Mossakowski, T.; and Codescu, M. 2016. Ontology Patterns with DOWL: The Case of Blending. In *Proc. of DL 2016*. Cape Town, South Africa: CEUR Vol. 1577.

Lakoff, G., and Johnson, M. 1980. *Metaphors We Live By*. University of Chicago Press.

Lakoff, G. 1987. *Women, Fire, and Dangerous Things. What Categories Reveal about the Mind*. The University of Chicago Press.

Mandler, J. M. 2004. *The Foundations of Mind : Origins of Conceptual Thought: Origins of Conceptual Though*. New York: Oxford University Press.

Neuhaus, F.; Kutz, O.; Codescu, M.; and Mossakowski, T. 2014. Fabricating Monsters is Hard - Towards the Automation of Conceptual Blending. In *Proceedings of C3GI*, 2–5.

Pereira, F. C., and Cardoso, A. 2003a. The horse-bird creature generation experiment. *AISB Journal* 1(3):369.

Pereira, F. C., and Cardoso, A. 2003b. Optimality Principles for Conceptual Blending: A First Computational Approach. *AISB Journal* 1(4).

Pereira, F. C., and Cardoso, A. 2006. Experiments with free concept generation in Divago. *Knowledge-Based Systems* 19(7):459–470.

Potsch, E., and Williams, R. F. 2012. Image schemas and conceptual metaphor in action comics. In *Linguistics and the Study of Comics*. Springer. 13–36.

Rosch, E. H. 1973. Natural categories. *Cognitive Psychology* 4(3):328–350.

Runco, M. A., and Jaeger, G. J. 2012. The standard definition of creativity. *Creativity Research Journal* 24(1):92–96.

Ruy, F. B.; Guizzardi, G.; de Almeida Falbo, R.; Reginato, C. C.; and Santos, V. A. 2017. From reference ontologies to ontology patterns and back. *Data Knowl. Eng.* 109:41–69.

Schmidt, M.; Krumnack, U.; Gust, H.; and Khnberger, K.-U. 2014. Heuristic-Driven Theory Projection: An Overview. In Prade, H., and Richard, G., eds., *Computational Approaches to Analogical Reasoning: Current Trends*, Computational Intelligence 548. Springer-Verlag.

Szawerna, M. 2012. Superpower corruption - the blended universe of Watchmen by Alan Moore, Dave Gibbons, and John Higgins. In Wasik, Z., and Post, M., eds., *Papers and Studies in Axiological Linguistics*. Philological School of Higher Education in Wrocaw Publishing. 87–101.

Investigating and Automating the Creative Act of Software Engineering

Simon Colton[1, 2], **Edward J. Powley**[1] and **Michael Cook**[1]
[1]The MetaMakers Institute, Falmouth University, UK
[2]Computational Creativity Group, Goldsmiths, University of London, UK
www.metamakersinstitute.com ccg.doc.gold.ac.uk

Abstract

We take the position that the creative act of computer programming has been under-investigated in Computational Creativity research. It is time for a concerted study of software engineering from the perspective of creative software. Such software should produce code and algorithms as artefacts of interest in their own right, rather than as a means to an end. We briefly survey and critique existing automated programming approaches, propose some novel methods for this, and investigate potential application areas for automated software engineering. Central to our position is the notion that creative software generators should perform in an unsupervised manner in order to problematise the world rather than (or in addition to) solving given problems. This will necessarily utilise some current methodologies and philosophies from Computational Creativity research, and we explore the ways in which these could guide future software synthesis approaches.

Introduction and Motivation

Creative people write software for a number of purposes. Often, coding is a means to an end of achieving some goal, automating some task or solving some problem. In these cases, the value of the written code is secondary to the value of running the code in a particular application. However, in other contexts, the software itself is appreciated as an important creation or discovery over and above any application of it (if there is one). As discussed below, examples of where this is the case include scientific discovery, automated creators, recreational coding, and games (for education or entertainment) which use coding as a game mechanic.

Like a product or process in the arts or sciences, code can take on a life of its own, being studied, modified, used in unforeseen applications and even celebrated culturally. We explore here the position that Computational Creativity research would be well served by thinking of computer programs as important artefacts in their own right, rather than purely as task-completing or problem-solving processes. We therefore advocate studying and automating the creative act of software engineering, similar to studying and automating the creative act of painting, writing, composing, etc.

It is tempting to point out that, as Computational Creativity researchers, we build software to generate artefacts such as paintings, poems, musical compositions, videogames, etc., and hence automating the engineering of such software systems would represent a meta-approach of value to the field. However, it is too ambitious currently to suggest the automatic production of generative systems in all but highly specialised applications. Moreover, this would obscure our point that the code artefacts themselves, rather than outputs from running the code, should be the end goal for implementations which simulate creativity in programming.

Critical to our outlook for the study of automating creativity in programming is the notion that it should be used to **problematise the world** rather than (or in addition to) solving given problems. By *problematise*, we mean that the generated code exposes opportunities either for better understanding the world through problem solving (e.g., the code exposes an unexpected anomaly or hypothesis about a dataset), or application of the code in cultural contexts to change the world (e.g., the code can be used as a mechanic in a videogame). In both cases, it is important to note that the generated code needs to be appreciated as a cultural artefact in its own right, for its aesthetic and knowledge-enhancing qualities and not just for its utility, in order for the opportunities to be fully understood and exploited.

As a hypothetical example, imagine in the early days of computer graphics that an automated code synthesis system had output an image filtering algorithm which performed edge detection, i.e., similar to the invention by Canny (1986). Imagine further that this was a completely unexpected piece of code, i.e., the user had no idea in advance that edge detection was even something that software could do. In this hypothetical context, the discovery of this algorithm would lead to a series of problems going from **unknown unknowns**[1] to known unknowns. In other words, the new discovery does not immediately provide all the answers about this new domain, but it does expose which questions are interesting for further study. Such exposed questions would include: how does this algorithm work; how good is the edge detection and how do we measure this; what are the practical applications of this software; what are the artis-

[1]The term 'unknown unknowns' is originally attributed to psychologists Luft and Ingham in their development of the Johari window technique. However, it was brought to popular attention by US Secretary of Defense Donald Rumsfeld in 2002, with the statement: "We also know there are known unknowns; that is to say we know there are some things we do not know. But there are also unknown unknowns – the ones we don't know we don't know."

```
send(to, from, count)
register short *to, *from;
register count;
{
    register n=(count+7)/8;
    switch(count%8){
    case 0: do{ *to = *from++;
    case 7:      *to = *from++;
    case 6:      *to = *from++;
    case 5:      *to = *from++;
    case 4:      *to = *from++;
    case 3:      *to = *from++;
    case 2:      *to = *from++;
    case 1:      *to = *from++;
        }while(--n>0);
    }
}
```

Figure 1: Duff's device (Duff 1988).

tic affordances of this software; what are other approaches to edge detection, etc. Solving some of these problems will lead to better understanding the world, while others will lead to new practical opportunities. Moreover, studying the code as an important cultural artefact in its own right may itself lead to the generalisation and formalisation of the edge detection process found in the generated code in language independent terms. It is not too far fetched to make an analogy here with formative artworks by an artist – while better algorithms may follow, having the original algorithm in the world is important historically and culturally.

Currently, the responsibility for asking these follow-up questions falls to the people who set the code generation system in motion and who study its output (which assumes that the output is comprehensible to humans – by no means a given). It is interesting to imagine an AI system which is capable of appraising generated code in this way.

The edge detection example above considers software from a functional perspective: the interesting part is the operation the code performs. However, the same principle can apply to more abstract code structures, where the the code itself is interesting (within the application domain of software engineering rather than elsewhere). For an historical example, Duff's device (1988) is a clever abuse of the `switch-case` statement in C (see figure 1), which allows a loop to be partially unrolled (often resulting in faster execution) without introducing any restrictions on the number of iterations the loop may execute. As previously, imagine this had been discovered not by a software engineer at Lucasfilm, but by an automated programming system. It similarly exposes new problems: how does it work; is it general; how much efficiency gain does it yield? Solving these may lead to new discoveries by human or by computer, just as Duff's device inspired a particularly elegant implementation of the coroutine idiom in C, as described by Tatham (2000).

In both these examples, an obviously useful piece of software has highlighted many new and interesting problems. While automated programming clearly has problem-solving applications, we believe that applications which expose unknown unknowns are key to modelling and utilising creative behaviours, which in turn is essential to studying the full potential of automated approaches to code generation. Creative behaviours free the approach from difficult constraints, but in turn introduce a number of difficulties in execution, which are discussed below. We believe that creative automated coding approaches will be able to enhance the arts, lead to scientific breakthroughs and drive progress in society. We further believe that advances in automated programming have been held back by the almost-universal application of them within the problem-solving rather than artefact generation paradigm of AI. Bringing creative automated coding into Computational Creativity research would open new avenues of research, provide a step change in the value of artefacts being generated and unearth new application domains. We would expect to see more interesting and sophisticated processes being undertaken by software, advancing our understanding of what it means for software to be creative.

The rest of the paper is organised as follows. In the next section, we look at various ways in which programming has been automated, and provide a critique highlighting the inappropriateness of these methods with respect to creative affordances and cultural celebration of code. Following this, we suggest alternative approaches for the automatic generation of code within the problematising paradigm described above, and highlight some potential applications. We conclude by discussing how automated programming could be guided by modern Computational Creativity practice and in return enhance our philosophical understanding of software being creative, and describe future research directions to explore the creative potential of automated code generation.

Automated Programming Approaches

In common usage, the term 'automatic programming' refers to a range of techniques devised to enable people to program more efficiently, e.g., source code creation through templates within an IDE. Within Artificial Intelligence research, the notion of the fully automated construction of computer programs is to be found within the fields of machine learning, evolutionary programming and automated programming synthesis. We look at each here with respect to their suitability for problematising the world via valued code generation rather than problem solving.

In machine learning, the most obvious areas where automated programming is found are when the learned classifiers are explicitly code, as with Inductive Logic Programming (Muggleton and De Raedt 1994), i.e., where Prolog programs are learned for classification and prediction problems. However, *all* machine learning methods effectively learn representations that are easily interpretable as computer programs. Importantly, deep learning methods are currently being investigated as automated programming systems, with the learned networks examined as computer programs in addition to approximations of neural structures. For instance, deep learning luminary Yann LeCun has recently stated:

> "Deep Learning has outlived its usefulness as a buzz-phrase . . . Vive Differentiable Programming! . . . the important point is that people are now building a new kind of software by assembling networks of parameterized functional blocks and by training them from examples using some form of gradient-based optimization."
> *facebook.com/yann.lecun/posts/10155003011462143*

As an example of the power of deep learning for automated coding, we can turn again to graphics. Previously, in order to transfer an artist's style, or a particular texture, from one image to another – e.g., producing a pastiche of an artist's work by applying a filter to a given image – one had to write a bespoke program or devise a macro in an application like Adobe's Photoshop. The graphics community has investigated style transfer for aspects such as colour (Abadpour and Kasaei 2007), and produced scores of individual style transfer methods for various artists/textures, with pastiche generation finding its way into mainstream graphics packages. However, this was only done for famous artists or particularly useful textures, as hand-programming style transfer methods was time consuming. With the advent of neural network approaches (Gatys, Ecker, and Bethge 2016), a deep neural model can be trained which applies the style of one image to the content of a second image, with impressive results and no user programming required. It is clear that each application of the approach generates a program, albeit in the form of a generative neural network, which performs the same function as the previously hand-crafted ones.

The drawbacks to the usage of machine learning techniques to expose truly unknown unknowns revolve around both the supervised nature of the application and the format of the programs which can be learned. In general, supervised machine learning techniques involve labeling examples into classes and generating methods which can classify unseen examples with high accuracy. As such, supervised methods are suitable for solving known unknowns, where the user knows what he/she wants, but doesn't know exactly what it looks like. However, this is too restrictive to imagine supervised machine learning approaches being used to expose unknown unknowns. Unsupervised methods like data mining can make discoveries the user didn't know they were looking for, but even in these situations, the nature of the discoveries is usually prescribed in advance, in terms of their structure, underlying concepts, or the way they relate variables in the data. Hence, these are not truly unknown unknown problems which are unearthed. If we refer back to the edge detection hypothetical example, that kind of discovery is of software which does something that no-one has thought of doing before, which does not project well onto either supervised or unsupervised machine learning methods.

Another drawback of deep learning techniques for our purposes is that their representation of a "program" is rather opaque from a human perspective. They do not produce code, at least not in a form that a human software engineer would recognise. A trained Artificial Neural Network (ANN) consists of a network structure (which is generally a product of human effort rather than of the AI system), along with thousands or millions of parameter values. Despite efforts to explain and visualise the workings of ANNs (Montavon, Samek, and Müller 2018), it is difficult to appreciate the beauty in a well-tuned ANN in the same way one might appreciate the beauty in a well-written program. For humans at least, it is easier to understand and appreciate a million lines of C++ code than to understand a million real-valued parameters. Notwithstanding efforts in Computational Creativity research to provide alternative scenarios in which creative software can be evaluated, e.g., modeling empowerment for intrinsic motivation (Guckelsberger, Salge, and Colton 2017) or societal curiosity (Saunders 2007), the evaluation of the products and processes of creative systems is largely human-centric. Hence, at least for the time being, it seems that if generated code is to be appreciated culturally, then it should either be understandable by humans (which, as argued below, most generated code is not), or there should be some way of generating high-fidelity explanations of it automatically, noting that different types of users will appreciate code in different ways (Cook et al. 2013).

Evolutionary programming techniques such as genetic programming (GP) produce code directly in a variety of languages, using a sophisticated array of search techniques with crossover and mutation at their heart, guided by user given (or sometimes machine learned) fitness functions. Moreover, they are used for both supervised tasks, e.g., generating classifiers for machine learning applications, and in unsupervised tasks, for instance in evolutionary art projects where the user specifies the fitness of genomes (code) by selecting between phenomes (e.g., images) generated by executing the code (Romero and Machado 2007). The expressivity of many GP approaches means that they could in principle construct code of real value which does something that no one has thought of doing before, potentially exposing unknown unknowns and problematising the world.

Unfortunately, the nature of crossover and mutation does not lend itself to the production of easily understandable code. Overly complicated code is not a problem when the value of the application of the software outweighs the value of the code itself. However, in projects where the code itself is to be celebrated, this issue could be a barrier to usage of GP approaches. In addition, there has been some recognition in the field of Computational Creativity, that the process by which an artefact is created is used in assessing the value of the artefact itself (Colton 2008), and that creative systems should *frame* both their processes and products in order to enable full appreciation of the creative act(s) they perform (Charnley, Pease, and Colton 2012). Artists have embraced the idea of Darwinian-like evolution of software driving artistic projects: they often use scientific descriptions of evolution when framing their art, and present the evolution of their pieces in terms of the family tree of offspring phenomes (e.g., images). However, neither of these is the same as using the actual construction process of a program to add value to the creative act and the product.

In a comprehensive survey, Gulwani et. al (2017) describe automated program synthesis as:

> ". . . automatically finding a program in the underlying programming language that satisfies the user intent expressed in the form of some specification."

With deep learning being used directly for program synthesis, as described by Balog et al. (2017), it seems likely that there will be a step change in the abilities of software to automatically generate code in this context. However, the situation in automated program synthesis is that – almost without exception – automatic generation of software is done to solve a particular problem in a supervised manner. Problem types include finding code which can turn given inputs into given outputs and improving existing code, for instance via genetic improvement as per the Gen-O-Fix software (Swan, Epitropakis, and Woodward 2014) or via code transplantation (Barr et al. 2015). While these approaches create new code, they do not cater for the situation where the user has data or existing code that he/she wants to investigate by automatically generating programs, but doesn't know exactly how that investigation should proceed. Indeed, to the best of our knowledge, it seems that no one has ever applied program synthesis in a setting where what constitutes a "good" generated program is unknown in advance and is to be discovered via the process itself. As such, it is difficult to imagine employing the methods developed in this field to generate code that no one knows anything about in advance.

Certain methods and methodologies from Artificial Intelligence research have found natural application in Computational Creativity projects, while others have barely been used. Evolutionary programming, for example, is a mainstay of the field, while machine learning has had fewer, but notable applications, and automated program synthesis approaches have – to the best of our knowledge – never been applied to tasks associated with creative behaviour. There have been successful ad-hoc approaches to direct code generation where, to some extent, the code has been appreciated over and above its value in application (Cook et al. 2013). Here, within the context of the generation of entire videogames by the ANGELINA system (Cook, Colton, and Gow 2016), code was generated directly to control the action of the player's character when they pressed the special powers key (the space bar) on a keyboard. Game levels were then generated around this special character function, i.e., which required the usage of the function in order to complete the level. The generated code was not simply applied, but examined as a valuable artefact, to pass on information to players about what it did (although this was not needed in most cases). Notwithstanding these ad-hoc approaches, in general the idea of creatively generating code as artefacts to be appreciated in their own right has been largely under-investigated in Computational Creativity research.

Creative Approaches to Automated Programming

Taking the above critique into account, to maximise the potential for automated code generation, we advocate approaches which produce **human-understandable** code in **human-like** ways. By "human-like ways" we mean mirroring the various ways in which human software engineers approach the task of programming: a set of logical iterative steps, top-down or bottom-up, drawing heavily on design patterns and other accepted wisdom, and generally bearing almost no resemblance to the search-based techniques often used by automated systems. In this way, the generated code could be properly appreciated by people, and the creative system can appeal to the code construction method when framing its efforts. We propose two such approaches here.

Given the logical nature of programming languages and the plethora of mathematical/statistical applications of code, it is not surprising that many early (and indeed many modern) computer scientists were originally mathematicians. In this vein, programming languages and logic have much overlap, with Prolog, for instance, being described as a form of logic, a database and a programming language, and logic and maths topics being essential in a computer science education. It is therefore not too difficult to imagine a generative system able to produce mathematical theories graduating from a mathematical discovery system to an automated coding system, which is precisely what we have started to do with the HR program (Colton and Muggleton 2006).

The latest iteration of the software, called HR3 and described by Colton, Ramezani, and Llano (2014), was re-engineered from scratch to be an automated programming system, while inheriting the mathematical abilities developed for previous versions. As an example of the difference in approaches, in the domain of number theory, HR2 was given background knowledge including data about which numbers wholly divide which other numbers, e.g., it was given the full set of divisors for the numbers 1 to 1000. It would then invent mathematical concepts, such as perfect squares and prime numbers, and express these in human-readable ways, e.g., as LaTeX sentences or in first order logic (for integration with third party automated theorem provers, model generators and constraint solvers). It would further hypothesise, using standard mathematical symbols, certain conjectures about the concepts which were empirically true, e.g., that an integer can never be both prime and square. Where possible, it would appeal to theorem provers and model generators to prove/disprove the conjectures, and use this to assess and rank the concepts and conjectures, in advance of presentation to the user.

In contrast, HR3 is given as background knowledge Java code which can generate integers up to a user-given limit on the number line, and code which can determine the divisors of a given integer. It then invents concepts in similar ways to HR2, but the concepts are themselves expressed as Java methods which can be run independently of HR3, with the user supplying integer inputs. Conjectures are similarly presented as Java methods which, when executed, test the conjecture empirically over given data (which, as in the case of number theory, can be generated). Hence, if a particular conjecture has tested true on the integers from 1 to a million before being presented, a user can choose to run the method up to 1 billion, before further investigation. The user can, of course, interpret the concepts and conjectures expressed as methods in Java in more mathematical ways if they so desire, but they are spared the common task of implementing them, as they are originally generated directly as code.

In addition to working with background knowledge that is Java code, HR3 can work with standard data from machine learning applications and/or Prolog files. In addition to the code-centric redesign of the software, another important innovation is the usage of *randomised data* to filter uninteresting conjectures. As described by Colton, Ramezani, and Llano (2014), HR3 is orders of magnitude faster than HR2 and is able to generate millions of concepts and conjectures in minutes/hours (depending on the extent of the data over which it is running). The space of Java methods which

HR3 can generate includes numerous duplicates, which on the surface look different, but perform essentially the same calculation. When HR3 makes conjectures relating these methods, they are ultimately disappointing, as they express an artefact of the code construction process rather than a discovery about the data over which the code is run. However, such disappointing conjectures can be discarded if they are also true of randomised data, as the probability of the conjecture arising because of a pattern in random data is miniscule, so the conjectures must relate code not data, and can be discarded (although HR3 can use the information about related code to substitute slower programs with faster ones). A final innovation in HR3 is an ability to make conjectures which are not 100% true, in preparation for working on noisy data.

We note that the HR systems have been used many times to problematise the world, by inventing mathematical concepts and asking questions about their value, then making conjectures and asking questions about their truth. As an early example, HR1 invented the concept of refactorable numbers as those integers for which the number of divisors is itself a divisor. It further hypothesised that odd refactorable numbers are square numbers, which – along with other generated conjectures – was proved by Colton (1999). This discovery was certainly an unknown unknown as the user (first author) had no conception of refactorable numbers in advance: HR1 told him something new and interesting in number theory that he didn't know he was looking for.

Many approaches to automated programming treat the problem as one of search: there exists some notional space of fully-formed programs, and the job of the automated programming system is to locate a particular point in this space. In contrast, programming as a human activity is a highly iterative process, in which a program is built and refined over a period of development. Arguably, this approach bears some resemblance to local search in the program space, but this does not seem to capture the nature of software engineering as an iterative process. There are many formalisms for iterative software engineering by people. An interesting example is test-driven development (TDD) (Beck 2002), which breaks programming into a series of rapid cycles. A new feature (or bug fix) is implemented into a program by (1) writing one or more unit tests to verify the prospective feature; (2) writing the bare minimum of code to cause the new test case to pass without breaking any existing tests; and (3) refactoring the code to add structure and remove duplication. TDD leads to better quality code at the expense of increased development time (George and Williams 2004). TDD could be used to guide automated programming, allowing systems to build a software artefact iteratively rather than monolithically, but to our knowledge, this has not been studied.

The game designer Sid Meier says that "[playing] a game is a series of interesting choices" (Rollings & Morris 2000). This also fits creative processes for artefact generation, so it seems promising to apply decision-making techniques from games in areas where a series of interesting choices must be made, which of course characterises programming. One such technique is Monte Carlo Tree Search (MCTS) (Kocsis and Szepesvári 2006; Chaslot et al. 2006), which is successful in a wide variety of adversarial games (Browne et al. 2012) and decision problems beyond games (Mańdziuk 2018). A closely related technique is Nested Monte Carlo Search (NMCS) (Cazenave 2009), which outperforms MCTS in deterministic single-agent domains. White, Yoo, and Singer (2015) compare MCTS and NMCS to more traditional Genetic Programming approaches. MCTS and NMCS are found to be competitive on some benchmark tests, which were rooted in the idea of automated programming as problem solving. Game tree search, as performed by MCTS approaches is objective-focused, in that it aims to "win the game" (i.e., find a terminal state with a high value according to some utility function). This lends itself more to the problem-solving paradigm of AI than to artefact generation. Methods such as novelty search (Lehman and Stanley 2011) or surprise search (Yannakakis and Liapis 2016), which explore a space for novel or surprising instances rather than seeking a global optimum, may be useful if the aim is to generate interesting programs rather than solve a concrete problem.

Potential Applications of Code Creation

We expect that giving software the ability to creatively generate code will have myriad uses. We propose the following general areas in which code-centric creative automated program generation may be employed with a significant impact.

- Problematising emerging scientific domains

Scientific understanding is constantly being updated in response to new results, which come from new data derived from new experiments, often via new machinery. On the cutting edge of scientific domains, breakthroughs such as improved instrumentation, a theoretical advance, or unexpected experimental results can lead to an explosion of activity when lots of concepts and conjectures are proposed and understanding emerges. As an example, brain scanning equipment occasionally becomes more accurate in sensing structure and activity in brains, i.e., at previously unseen resolutions. In response, physical models of the brain can be challenged and updated and/or new ones invented to capture the information arising from the higher resolution scans.

In these emerging fields, there is as much a need to problematise current understanding as there is to solve problems which have arisen. When people do this, they notice patterns with no explanation, invent concepts to capture groupings without knowing precisely the conceptual definition, pose hypotheses on small amounts of empirical evidence and attempt to find more substantial support, and attempt to derive explanations to phenomena without the necessary language. Often, as the concepts, conjectures and explanations become more concrete, they will be turned into program code to become operationalised, which affords more accurate study of the scientific data being harvested.

We propose to automate the task of problematising scientific understanding from the opposite direction. That is, rather than starting from observations and ending up at code, we suggest using a system such as HR3 to start by automatically inventing code which exposes previously unforseen patterns in data, then conceptualising from the code. In particular, we initially intend to frame the task of problematising a given dataset as finding quadruples of code $\langle A_1, A_2, A_3, A_4 \rangle$. Here, algorithms A_1 and A_2 manipulate data to produce outputs related by algorithm A_3, and algorithm A_4 shows that this relationship may be interesting. As an example, A_1 and A_2 could output a number for each datum in the dataset, A_3 could relate A_1 and A_2 with a boolean output which is true whenever A_2 produces a multiple of A_3, and A_4 measures the proportion of data points for which this is true. An individual quadruple could be selected and presented because, for the relationship A_3, the output of A_4 was the highest among those with A_3 in the third slot.

The algorithms in any quadruplet which is statistically significantly true of the data can be analysed as an empirically-supported hypothesis, which may lead to more general conceptual definitions, from which more hypotheses will flow, leading ultimately to an improved theoretical understanding of the processes which produced the data, via the lens of the generated algorithms. By enabling the automatic generation of all four algorithms in a quadruple, we hope to maximise the chances of discovering problems that are both interesting and truly unexpected. It may even be feasible and desirable to close the loop on the scientific process and automate the design and execution of experiments to gather more data in response to the discovery of patterns expressed via generated code, as per the *Robot Scientist* described by Sparkes et al. (2010), where machine learning, rather than creative code generation, drove the process.

- Self-modifying automated creators

Many existing creative systems could be enhanced by enabling the software an to alter its own code and/or produce new code for enhanced functionality. We hypothesise that this would lead to increased appearance of autonomy, surprising levels of novelty and more sophisticated processing in the creative systems. As an example, we investigated painting style invention in Colton et al. (2015). This was done with offline generation of different styles as sets of parameters, with online style choices driven by machine vision. An alternative approach would be for the software to invent code to control aspects of the painting process, e.g., how it simulates natural media, and how it uses them in simulated drawing and painting processes. In this context, we can imagine generative software detailing the code generation in commentaries to accompany paintings, framing its creations with descriptions of how it invented new processes – which could have much cultural appeal in the arts.

The FloWr system has been used for process invention, via the generation of novel flowcharts which act as generative text systems, as described by Charnley, Colton, and Llano (2014). As an example output, FloWr invented a flowchart which took the speeches of Churchill as input, extracted phrases with high negative or positive valence, then outputs them in pairs where each has the same footprint (number of syllables). This approach could be enhanced with code invention for the processes *inside* each node of the flowchart. One cultural application of this that we plan to undertake is to produce an anthology of poems, each of which has been generated by a different flowchart, with nodes containing generated code. Here, each poem would be portrayed alongside the code which generated it, so both can be celebrated in the anthology, similar to how Montfort and Fedorova (2013) present both poems and code.

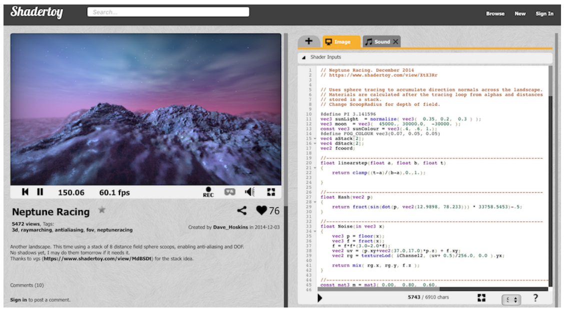


Figure 2: Example ShaderToy post, where users share WebGL shader programs producing 3D animated effects (in this case, a flyover of a procedurally generated landscape).

As mentioned previously, Cook et al. (2013) investigated how direct code generation could be used to successfully invent game mechanics for videogames. Here, the ANGELINA system performed code generation as part of the game design process, but this could be taken further, so that games themselves procedurally generate code for game mechanics, in a similar way to how they perform procedural content generation (PCG). Just as PCG keeps games fresh, extends their lifetime and adds intrigue, procedural mechanic generation could do similar, perhaps in a puzzle context, where players have to work out what the game mechanic is doing through usage of it. In this context, we can imagine the game presenting the code for a game mechanic in user-understandable ways, as hints to how best to use it.

- Contributing to recreational coding communities

There are several communities of programmers who write code for recreational, rather than pragmatic, purposes. These have varying levels of application to "real-world" domains. At one end of the scale, the *ShaderToy* community (shadertoy.com), illustrated in figure 2, write GPU fragment shader programs to produce complex 3D visualisations, which have applications in the development of games and other real-time graphics. ShaderToy's roots can be traced back to the *demoscene*, a digital art culture which arose in the 1980s around producing advanced graphical and audio effects from the home computers of the time (Reunanen 2017). Many of the techniques developed by demoscene coders, particularly in the domain of real-time 3D rendering, found applications in games and other software domains.

Much less concerned with real-world applicability are those programmers who write *quines* (Hofstadter 1979), which are programs that output their own source code, and *polyglots*, which are programs whose source code is a valid program in multiple languages, performing the same or different tasks in each. These exercises in creative intellect have arguably no practical purpose, but the activity of programming and reading the programs of others is an interesting pastime to many. An even more extreme example is the *International Obfuscated C Code Contest* (ioccc.org), where

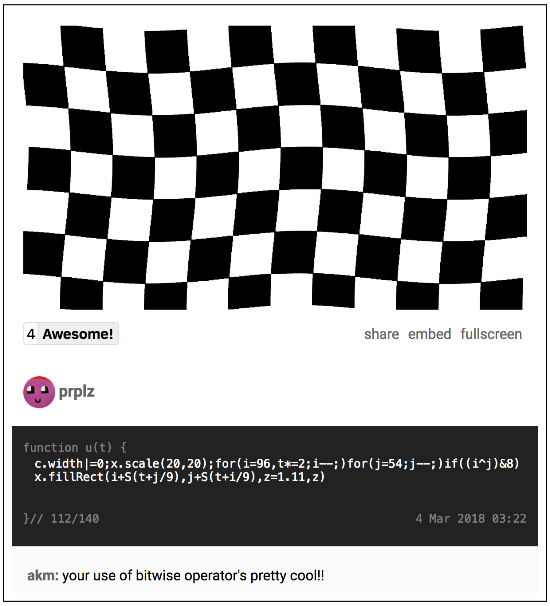


Figure 3: Example Dwitter post (140-character JavaScript program producing animated effects) of a chequered flag blowing in the wind, and follower comment.

participants compete to solve problems in the most unusual and intentionally obtuse ways possible, contrary to the usual principles of good software engineering. Between the two extremes are communities such as *Dwitter* (dwitter.net, see fig. 3), and *Code Golf* (codegolf.stackexchange.com), where programmers write programs (to generate animated graphics and solve various computational problems, respectively), but try to do so with the fewest characters possible.

The appeal of recreational programming is twofold. To the programmer, it is an enjoyable intellectual exercise, like a cryptic crossword or logic puzzle. In this regard, it has similarities to programming-based puzzle games such as *Shenzhen I/O*, *Silicon Zeroes* and *Human Resource Machine*, where players code as part of the gameplay, although these games tend to adapt a more traditional view of coding as problem solving, albeit in a game environment. To the community, it is a celebration of code as an artefact that exists solely for its own sake, to be appreciated in itself, rather than as producing useful/pleasing output. We believe that automated creative programming approaches such as those described above could contribute to recreational programming communities, perhaps producing intriguing software that people might not normally think of coding, in a similar way to how boardgame playing agents occasionally make moves that grandmasters would not necessarily think of.

Conclusions and Future Work

We have argued that Computational Creativity research would be well served by investigating the creative act of software engineering and implementing systems for generative coding. We proposed that such automated software production should lead to culturally appreciated programs which problematise the world, providing enhanced understanding and opportunities in areas like scientific discovery, generative systems and recreational coding. We proposed approaches including extending mathematical theory formation to automated programming, and applying MCTS to making decisions in iterative code formation.

Whatever the method for achieving creative automated programming, if applied for the purposes of problematising the world, this will need to be guided by current thinking in Computational Creativity research. The problem solving paradigm has dominated over the artefact generation paradigm in AI research largely because solving carefully hand crafted problems is guaranteed to be valuable, and systems can be formally evaluated in terms of quality of solution, efficiency, coverage, etc. Notwithstanding some competitions pitching one generative system against another, such as the Mario level generation competition (Shaker et al. 2011), it is generally difficult to compare and contrast the kinds of artistic creations or scientific discoveries that Computational Creativity systems generate. This will be exacerbated when generated code artefacts expose unknown unknowns, as the user should be totally unaware of the problem posed, and hence likely to not recognise its value easily.

In this context, it seems likely that the software will have to make some efforts to convince users of the value of the code it generates, drawing on approaches to framing its processes and products as per Charnley, Pease, and Colton (2012). Moreover, evaluations of generated algorithms may actually involve assessing the creativity of the software which produced them, and could draw on work by Colton, Pease, and Charnley (2011). In return, enabling software to generate code could help solve a philosophical issue which we could term the "mini-me" problem: that creativity is projected onto the programmer and/or the user of generative software because of the explicit nature of the instructions given through human programming. When creative software can in principle rewrite its entire code-base, it will be possible to argue that the programmer has negligible effect on how the software operates or what it produces.

The first steps towards applying the HR3 system to creative code generation have been taken (Colton, Ramezani, and Llano 2014), but there is much work still left to do. In particular, we need to address how to scale up from the software producing small programs to more sophisticated software. We will be investigating both the approach to producing code-quadruples to expose unknown unknowns described above, where four algorithms will be related, and HR3 inventing code to sit inside larger flowcharts generated by the FloWr system (Charnley, Colton, and Llano 2014). We will also look into how sampling methods for both data and the code space can be used to enable HR3 to search deeper for larger algorithmic constructions. Finally, we plan to develop a methodology whereby domain experts such as scientists or game designers can easily provide guidance to the code generation process in terms of mathematical/programmatic/logical ingredients for code and give feedback about the quality of the code produced.

The study of deep learning systems as generators of computer programs hints at an explosion of interest across computer science in software systems programming themselves and/or automatically designing code for practical and problem solving purposes. Within this ecosystem, there will be space for a range of approaches, including unsupervised approaches to creative automated program synthesis as proposed here. It is possible that deep learning will be seen historically as the first truly successful approach to automated programming, and we hope that more unsupervised approaches, driven by the desire to problematise the world through generated code celebrated in its own right, will also be influential in the progress of computer science.

Acknowledgements

We would like to thank the anonymous reviewers for their insightful suggestions, which have improved this paper. This work has been funded by EC FP7 grant 621403 (GRO).

References

Abadpour, A., and Kasaei, S. 2007. An efficient PCA-based color transfer method. *J. Visual Comm. and Image Representation* 18(1).

Balog, M.; Gaunt, A.; Brockschmidt, M.; Nowozin, S.; and Tarlow, D. 2017. Deepcoder: Learning to write programs. In *Proceedings of the International Conference on Learning Representations*.

Barr, E.; Harman, M.; Jia, Y.; Marginean, A.; and Petke, J. 2015. Automated software transplantation. In *Proceedings of the International Symposium on Software Testing and Analysis*.

Beck, K. 2002. *Test-Driven Development*. Addison-Wesley.

Browne, C.; Powley, E. J.; Whitehouse, D.; Lucas, S. M.; Cowling, P. I.; Rohlfshagen, P.; Tavener, S.; Perez, D.; Samothrakis, S.; and Colton, S. 2012. A Survey of Monte Carlo Tree Search Methods. *IEEE Trans. on Computational Intelligence and AI in Games* 4(1).

Canny, J. 1986. A computational approach to edge detection. *IEEE Trans. Pattern Analysis and Machine Intelligence* 8(6).

Cazenave, T. 2009. Nested Monte-Carlo Search. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence*.

Charnley, J.; Colton, S.; and Llano, T. 2014. The FloWr framework. In *Proc. 5th Int. Conference on Computational Creativity*.

Charnley, J.; Pease, A.; and Colton, S. 2012. On the notion of framing in computational creativity. In *Proceedings of the 3rd ICCC*.

Chaslot, G; Saito, J.; Bouzy, B.; Uiterwijk, J.; and van den Herik, H. 2006. Monte-Carlo Strategies for Computer Go. In *Proceedings of BeNeLux Conference on Artificial Intelligence*, 83–91.

Colton, S., and Muggleton, S. 2006. Mathematical applications of Inductive Logic Programming. *Machine Learning* 64.

Colton, S.; Halskov, J.; Ventura, D.; Gouldstone, I.; Cook, M.; and Perez Férrer, B. 2015. The Painting Fool sees! New projects with the automated painter. In *Proceedings of the 6th International Conference on Computational Creativity*.

Colton, S.; Pease, A.; and Charnley, J. 2011. Computational creativity theory: The FACE and IDEA descriptive models. In *Proc. of the 2nd International Conference on Computational Creativity*.

Colton, S.; Ramezani, R.; and Llano, T. 2014. The HR3 discovery system: Design decisions and implementation details. In *Proceedings of the AISB Symposium on Scientific Discovery*.

Colton, S. 1999. Refactorable numbers - a machine invention. *Journal of Integer Sequences* 2.

Colton, S. 2008. Creativity vs. the perception of creativity in computational systems. In *Proc. AAAI Spring Symp. Creative Systems*.

Cook, M.; Colton, S.; Raad, A.; and Gow, J. 2013. Mechanic miner: Reflection-driven game mechanic discovery and level design. In *Proceedings of the EvoGames Workshop*.

Cook, M.; Colton, S.; and Gow, J. 2016. The ANGELINA videogame design system, parts I and II. *IEEE Transactions on Computational Intelligence and AI in Games* 9(2) and 9(3).

Duff, T. 1988. Re: Explanation, please! `http://www.lysator.liu.se/c/duffs-device.html`.

Gatys, L.; Ecker, A.; and Bethge, M. 2016. Image style transfer using convolutional neural networks. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*.

George, B., and Williams, L. 2004. A structured experiment of test-driven development. *Information and Software Technology* 46.

Guckelsberger, C.; Salge, C.; and Colton, S. 2017. Addressing the 'Why?' in computational creativity. In *Proceedings of the 8th International Conference on Computational Creativity*.

Gulwani, S.; Polozov, O.; and Singh, R. 2017. Program synthesis. *Foundations and Trends in Programming Languages* 4(1-2).

Hofstadter, D. R. 1979. *Gödel, Escher, Bach: an Eternal Golden Braid*. Basic Books.

Kocsis, L., and Szepesvári, C. 2006. Bandit based Monte-Carlo Planning. In *Proc. of European Conference on Machine Learning*.

Lehman, J., and Stanley, K. 2011. Abandoning objectives: Evolution through the search for novelty alone. *Evo. Comp.* 19(2).

Mańdziuk, J. 2018. MCTS/UCT in solving real-life problems. In *Advances in Data Analysis with Computational Intelligence Methods*. Springer.

Montavon, G.; Samek, W.; and Müller, K.-R. 2018. Methods for interpreting and understanding deep neural networks. *Digital Signal Processing* 73.

Montfort, N., and Fedorova, N. 2013. Small-scale systems and computational creativity. In *Proceedings of the 4th International Conference on Computational Creativity*.

Muggleton, S., and De Raedt, L. 1994. Inductive Logic Programming: Theory and methods. *Logic Programming* 19-20(2).

Reunanen, M. 2017. *Times of Change in the Demoscene: A Creative Community and its Relationship with Technology*. Ph.D. Dissertation, University of Turku.

Rollings, A., and Morris, D. 2000. *Game Architecture and Design*. Coriolis.

Romero, J., and Machado, P., eds. 2007. *The Art of Artificial Evolution: A Handbook on Evolutionary Art and Music*. Springer.

Saunders, R. 2007. Towards a computational model of creative societies using curious design agents. In *Engineering Societies in the Agents World VII, Vol. 4457 of LNAI, Springer*.

Shaker, N.; Togelius, J.; Yannakakis, G.; Weber, B.; Shimizu, T.; Hashiyama, T.; Sorenson, N.; Pasquier, P.; Mawhorter, P.; Takahashi, P.; Smith, G.; and Baumgarten, R. 2011. The 2010 Mario AI championship: Level generation track. *IEEE TCAIG* 3(4).

Sparkes, A.; Aubrey, W.; Byrne, E.; Clare, A.; Khan, M.; Liakata, M.; Markham, M.; Rowland, J.; Soldatova, L.; Whelan, K.; Young, M.; and King, R. 2010. Towards robot scientists for autonomous scientific discovery. *Automated Experimentation* 2(1).

Swan, J.; Epitropakis, M.; and Woodward, J. Gen-O-Fix: An embeddable framework for dynamic adaptive genetic improvement programming. *Technical report CSM-195, University of Stirling.*

Tatham, S. 2000. Coroutines in C. `https://www.chiark.greenend.org.uk/~sgtatham/coroutines.html`.

White, D. R.; Yoo, S.; and Singer, J. 2015. The programming game: Evaluating MCTS as an alternative to GP for symbolic regression. In *Proc. Genetic and Evo. Computation Conference*.

Yannakakis, G. N., and Liapis, A. 2016. Searching for surprise. In *Proc. 7th International Conference on Computational Creativity*.

Targeted Storyfying: Creating Stories About Particular Events

Pablo Gervás
Facultad de Informática
Instituto de Tecnología del Conocimiento
Universidad Complutense de Madrid
pgervas@ucm.es

Abstract

The present paper proposes a computational model of the task of building a story from a set of events that have been observed in the world. For the purposes of the paper, a *story* is considered to be a particular type of sequential discourse, that includes a beginning, a complication and a resolution, concerns a character that can be clearly identified as a protagonist, and ends with a certain sense of closure. Starting from prior approaches to this task, the paper addresses the problem of how to target particular events to act as the core of the desired story. Two different heuristics – imaginative interpretation and imaginative enrichment – are proposed, one favouring faithful rendering of the observed events and the other favouring strong cohesive plots. The heuristics are tested over a simple case study based on finding interesting plots to tell inspired by the movements of pieces in a chess game.

Introduction

Interest in stories as means of communication has risen over the past decade. As a result research efforts targeting the computational treatment of stories have proliferated. The academic world is beginning to realise that there is much more to how we use stories in communication than just understanding them or generating them. People interact with stories in many different ways, and in fact most of the situations in which we create stories in our everyday life do not involve inventing the events that make up the story. Rather, they involve constructing a story out of events that we are aware of, in order to communicate either the events themselves or some particular interpretation of them. Yet sometimes, for the sake of better communication, we do invent some events to give the story cohesion, or charm. This task has been overlooked in computational approaches to stories, in favour of more creative approaches to storytelling, where the complete story – including its constituent events – is made up from scratch. Yet there is special kind of creativity involved in coming up with a story to match a given set of facts. As we know, finding "the right story" to tell about a set of facts can be crucial for successful communication, making the difference between flat rendition of the facts and either an entertaining yarn or a convincing argument. The development of computational models of this task would be an important contribution to the field of computational narratology.

Although the task of composing a narrative based on a given set of events that have taken place is very relevant to understand how humans structure their perception and memory of the world around them, it has received less attention in terms of computational modelling than other tasks related to stories (story understanding (Mueller 2003) or story generation (Gervás 2009; Kybartas and Bidarra 2016)).

Part of the problem is that the study of this task from a computational point of view requires a model of the input (the set of facts observed / remembered that constitute the source and the starting point for the composition). Such a representation of the input is implicit in the specification of the task, yet any attempt at computational modelling must start by representing it explicitly, as it will significantly influence the rest of the process.

Advances have been made in the understanding of the task by considering a chess game as a very simple model of a formalised set of events susceptible of story-like interpretations. Chess provides a finite set of characters (pieces), a schematical representation of space (the board) and time (progressive turns), and a very restricted set of possible actions. In this approach, each individual chess piece taking part in the game is considered a character. Perception range is defined as the small space of N x N squares of the boad that constitutes that immediate surroundings of each piece at any given moment. Events are triggered by pieces moves. Whenever a piece moves, this constitutes an event for the piece itself, for any other piece captured during the move, and for any other piece that sees either the full move, the start of the move or the conclusion of the move. Fibres for each of the pieces are built by collecting event descriptions for those moves that they are involved in or they see. The same event may get described differently in different fibres depending on the extent to which the corresponding focalizer is involved in it.

The use of chess game data as a valid test domain relies on the assumption that these games are being interpreted as summaries of the movements and interactions of people over a map of space. It is important to note that intuitions arising from the rules of chess or chess playing experience must be disregarded for the approach to be successful. If this is achieved, the mechanics of narrative composition developed for this type of example may provide a valid source for extrapolation to more complex domains.

Related Work

The work in this paper is informed by existing theories of narrative, and some fundamental concepts of narratology need to be taken into account. The paper addresses the task of storyfication, which involves building a narrative construct from a set of events that have been observed in the world but do not necessarily fit together as a story before being processed. It is related to the task of narrative composition, which involves elaborating a set of observed events into a sequential discourse to be told. Although the difference is subtle, we assume that the result of a process of narrative composition is a sequential discourse that conveys a set of events, and the result of a process of storyfication is a particular type of sequential discourse, that includes a beginning, a complication and a resolution, concerns a character that can be clearly identified as a protagonist, and ends with a certain sense of closure.

Narrative

Narrative has been classed among the elementary cognitive abilities exhibited by human beings (Schank and Abelson 1977; Bruner 1991; Herman 2004). In particular, it is known to be the process by which humans transform a particular experience of reality into a cognitive form that is easy for the mind to store and to be communicated to other people. Based on these ideas, recent years have seen a significant effort to relate narrative to the study of human cognition (Herman 2003; 2013). An important obstacle that faces this challenge is the fact that humans are notoriously poor at identifying the processes that they apply in processing reality (Nisbett and Wilson 1977). The underlying latent processes have to be postulated from the observation of their external manifestations, such as the actual narratives as literary works –studied by narratology – or the processes by which humans produce narratives – studied by cognitive science.

Relevant concepts from the field of narratology (Abbott 2008) are the distinction between *fabula* – the set of events behind a story, independently of how they are rendered – and *discourse* – the particular way of rendering a given fabula as a sequence of statements –, and *focalization* (Genette 1980) – the way in which a story is told from the view point of particular characters, switching between them at need to cover what happens elsewhere.

Existing narratives can very rarely be paired with alternative records of the experience that led to them, or even the events that are represented in them. This is a significant obstacle for applying a data-driven approach to model narrative construction computationally, as these approaches require instances of both the input that lead to the communication impulse, the narrative that arose from it, and possibly representations of intermediate design decisions.

Cognitive scientists have proposed models of the *writing task*. Flower and Hayes define a cognitive model of writing (Flower and Hayes 1981) in terms of three basic process: planning, translating these ideas into text, and reviewing the result with a view to improving it. These three processes are framed by "the rhetorical problem" – the rhetorical situation, the audience and the writer's goals. The target events considered in the present paper would be an instance of part of this problem.

Narrative Composition

Operating on simple representations of a chess game in algebraic notation, exploratory solutions for the tasks of content selection and content planning are explored based on a fitness function that aims to reflect some of the qualities that humans may value on a discourse representation of a story. Based on this approach prior work has been carried out on exploring computational models of the task of narrative composition as a set of operations that need to be carried out to obtain a span of narrative text from a set of events that inspire the narration (Gervás 2012; 2013; 2014).

Work has also been carried out on the composition of narrative discourse from generated plots represented as plans (Winer and Young 2016). Although such efforts are not grounded on a set of events that actually happened, their approach resembles the work presented here in that the planning stage that creates a plot involves selecting a subset of all possible events based on how they might be connected (in this case, via causality), and subsequent processes determine how the selected events are organised into a discourse.

Storyfying

A computational model of the task of storyfying has been proposed in the *StoryFire* application (Gervás 2018). This model is based on a series of stages:

1. establishing how the events are perceived from the point of view of the participating agents, by partitioning experience into narrative threads centred on particular characters (a task known in narratology as *focalisation* (Genette 1980))
2. representing the structure of the story (or plot) to be constructed as an abstract frame to which the perceived events must be matched
3. mapping the events in (possibly a select part of) the narrative thread for some character into an abstract frame for a plot
4. generating a readable version of the resulting discourse

The *StoryFire* application relies on the solution for focalisation presented in (Gervás 2012; 2014), which partitions the perception of the world by a given agent into a *fibre* constructed as a sequence of events descriptions. An *event description* consisting of a set of predicates that encode the elements that appear within the perception range of the agent at a given point in space and a given moment in time. An example of an event description is given in Table 1, showing what the left white rook (`lwr`) sees around itself (the first, second and third white pawns, the left white knight, and the left white bishop) on the seventh move of the game from position `a 1` and what it sees happening (the third white

```
Focalizer: lwr
Position: a 1
Time: 7
Perception Range: 2

DESCRIPTIVE:
  is_at(wp1, a2)
  is_at(wp2, b2)
  is_at(wp3, c2)
  is_at(lwk, b1)
  is_at(lwb, c1)

NARRATIVE:
  leaves_from(wp3, c2)
```

Table 1: Example of event description, which acts as the basic unit of description for a narrative fibre.

```
PLOT ELEMENT NAME = CoupleWantsToMarry
ROLE-DATA
lover hero
beloved love-interest
```

Table 2: Description of the CoupleWantsToMarry plot element.

pawn moves two squares forward, and thereby disappears from view).

That earlier version of the *StoryFire* application relied on a representation of plot in terms of *plot frames*, which are representations as sequences of character-function-like elements (Propp 1928) known as *plot elements*. Each plot element holds a label (such as `CoupleWantsToMarry`) and a mapping between roles relevant to the plot element (such as `lover` and `beloved`) and roles relevant to the plot in general (such as `hero` and `love-interest`). An example of plot element is shown in Table 2.

The plot frames considered for earlier version of the *StoryFire* application were instantiations of the seven basic plots defined by Booker (Booker 2004).

The actual storyfication process produced a *match* between a thread and a plot frame involves an allignment between a subset of the events in a thread and the sequence of plot elements in a plot frame (described in terms of which time points in the thread are alligned with which plot elements in the frame), a mapping between the characters present in the thread and the plot roles in the plot frame and a score that corresponds to the percentage of satisfaction of set of roles involved in the plot element by roles assigned to the characters present in the matched event, averaged over all the allignment. An example of such a mapping is given in Table 3.

The preceding version of the StoryFire application produced stories that could be considered narrative interpretations of particular threads from a chess game. Given a particular piece playing in the game, the application would produce a story plot that involved that piece as protagonist and which was actually a selection out of the set of events in the narrative thread experienced by that piece during the game.

This approach was sufficient for emulating the simpler

```
Thread lwk
PlotFrame Comedy-UnrelentingGuardian
Score 83

ALLIGNMENT
 9 [0]
11 [1]
16 [2]
17 [3]

MAPPING
bp4=love-interest
rwb=obstacle
lwk=hero
```

Table 3: Match between thread and plot frame

kind of storytelling that people apply, for instance, on returning from a trip. A story to tell, extracted from the events experienced during the trip, is sufficient. However, the present paper attempts to address a refined version of the task, which involves not just finding a story to tell about the trip, but finding a story that includes particular events that happened during the trip.

In addition, the set of plots considered in the earlier version was built of plots that were structurally very similar to one another. This restricted the sequences of events for which matches could be found.

Targeted Storyfying

Prior approaches to the task of narrative composition assumed that the goal was to obtain the best possible story for a given set of events. In the present paper, we want to narrow the focus to obtain stories that include a specific subset of events. In terms of the example used above, rather than build the best possible story out of the trip to a given conference, we want a story about the trip that involves, say, the keynote presentation at the conference, even if the trip might yield better stories by focusing on the conference dinner instead. This ability to drive the storyfication process towards particular events would bring the functionality being developed a step closer to human capabilities.

To this end, we need to address three different issues. First, we need to establish some means for specifying which events are to be considered as strictly required. This specification should be considered as an input in addition to the wider set of events to be considered. Second, the additional restriction imposed on the procedure may rule out matches with certain plots, and there is a risk that no match be found by applying the set of plots and the baseline algorithm previously available. Third, a procedure for guaranteeing that the produced stories include the desired events.

Establishing a Target Seed for the Storyfication

The type of constraint that humans considered when carrying out (the process that closely resembles what we are now calling) targeted storyfication is very broad. For instance, one may desire a story about a particular event, but may want the event to initiate the story, or to conclude it, or appear somewhere in the middle of it. Or one may want the story to take place at a particular location, or involve a given object. For the purpose of the present paper, we want to identify the simplest possible specification of these constraints that is compatible with the representation we are considering, and which satisfies the requirement of driving the process towards a particular subset of the input material.

The narratives we are considering are already focalised on particular characters. The simplest additional restriction that can be imposed is to consider as target a particular moment in time. This is consistent with the representation for a chess game, which is partitioned into a sequence of time points corresponding to alternating piece movements between black and white. It also allows isolation of particular events in terms of piece movements. Finally, given that the perception of the game is focalised on a particular character, specifying a moment in time also restricts the location to wherever the focaliser is at that moment.

We will therefore specify the target for our specification as a list of time points in the game. For simplicity, we will consider these time points in chronological order. This input we will refer to as the *target seed*, as the story to be built ought to be constructed around it.

Increasing the Range and Complexity of Possible Stories

The set of plots that had been considered in earlier attempts was grounded on existing accounts of plot, and it was reasonably varied in terms of the set of plot elements that it included, but proved to be ill suited to the task. First, because it included a number of classical plot structures that required that the hero travel away from home and then return. Such a structure leads to great stories, but it is very unlikely to occur in the context of a chess game. Second, because part of the plot elements included to add variation corresponded to Propp's Donor cycle, where the hero meets a character that gives to him a magical object which can then be used to solve difficulties later in the story. Again, in the context of a chess game, transfer of objects between characters (pieces) is not contemplated.

Solving these two problems was easy simply by eliminating from the set of plots those that involved journeys or donors. But as a result the set of plots was significantly impoverished, both in terms of number of plots and variety of plot element combinations. To address this problem, a refinement on the representation of plot was introduced.

The plots considered for the present paper are represented in terms of plot spans. A *plot span* represents a span of plot, constituted by a sequence of plot element (or smaller spans). The idea is to capture the concept of a number of plot elements appearing as a structural unit in a plot, but not necessarily occuring contiguously in the discourse for the plot. For example, a plot span representing an Abduction as it features in classic stories would include the actual kidnapping (which would happen somewhere towards the start of the story) and the corresponding Release (which would happen somewhere towards the end of the story), but these two plot element are structurally connected. Such cases we refer to as an *axis of interest*. Axes of interest can be combined together, weaving their corresponding subspans with those of other axes of interest, to form complex plots (which are themselves represented as plot spans). The set of plot structures described in the literature (Gervás, León, and Méndez 2015) can be represented with the help of these elements.

```
AXISofINTEREST = Abduction
PROTAGONIST = abducted
ROLES = abducted abductor rescuer

PLOT-SPAN-NAME = Kidnapping

PLOT ELEMENT NAME = Abduction
ROLE-DATA
abductor x
abducted y

PLOT-SPAN-NAME = Rescue

PLOT ELEMENT NAME = Rescue
ROLE-DATA
abducted y
rescuer z
```

Table 4: The Axis of Interest for Abduction

```
PLOT-SCHEMA = OCM-Abd
PROTAGONIST = hero

Abduction Kidnapping  (abductor=villain,abducted=victim)
CallToActionReward Call (called=hero,caller=sender)

Abduction Rescue (abducted=victim,rescuer=hero)
CallToActionReward Reward (rewarded=hero)
```

Table 5: Example of plot schema for a basic Abduction plot

An example of axis of interest is shown in Table 4. To assist in the process of combining them into more elaborate structures, each axis of interest specifies which character is the protagonist and what the roles relevant to the axis of interest are.

Axes of interest are combined into plots by means of plot schemas. A *plot schema* encodes the way in which several axes of interest combine together to form the plot span for an elaborate plot. An example of plot schema is presented in Table 5.

This shows how the Abduction and CallToActionReward axes of interest are interleaved to form the basic plot, and how the narrative roles for the plot (hero, villain, victim, sender) are mapped to the roles specific to the constituent plot elements (abductor, abducted, called, caller, rescuer, rewarded). This information is necessary to ensure that, once characters extracted from the observed set of events are mapped onto to the set of narrative roles for the plot, coherent instantiation of the plot elements with the given characters can be carried out.

The simplicity of schemas allows for the rapid construction of a large number of variations of simple plots by combining a reduced set of axes of interest, while allowing for significant structural complexity in the resulting plots, arising from the interleaving of the axes of interest.

Creating Stories for Particular Fragments of a Chess Game

The procedure to be applied for targeted storyfication is an extension of the procedure applied in the StoryFire application as reported in (Gervás 2018). That procedure involved traversing the search space of pairings between the given thread and each of the candidate plot structures. Although these plot structures are now internally represented as plot spans rather than plot frames (see sections above on *Storifying* and *Increasing the Range and Complexity of Possible Stories* for details on the differences), at the time of computating the allignments they are still converted into a sequence of plot elements to simplify the computation.[1]

For each pairing between a thread and a plot, the procedure extracts the set of characters in the thread and the set of narrative roles to be filled in the plot, and considers all possible mappings between these two sets. For each such mapping, the procedure heuristically explores possible allignments between a subset of the event descriptions in the thread and the plot elements in the plot. Any such allignment respects the relative order of events in the thread and plot elements in the plot and provides a correspondence between some of the events in the thread and each of the plot elements in the plot. Each pairing between and event description and a plot element is scored in terms of the percentage of satisfaction of set of roles involved in the plot element actually assigned to the characters present in the event. The allignments themselves are scored in terms of the average of the scores for the pairing made for all their plot elements.

For each candidate plot, only the best scoring allignment is considered, under the assumption that once a good story has been constructed out of a given thread, additional stories from that same thread – with the same plot but a different cast of characters and/or differnet allignment with the events in the thread, and with lower scores – are not desirable.

To implement the targeting of a specific subset of the input thread, two possible heuristics are applied. The first heuristic attempts to emulate the behaviour of a person trying to tell a story about some events in her day, but committed to being strictly truthful about it. It involves search for a story constructed entirely out of events that did happen. We refer to this approach as *imaginative interpretation*, because it is built up entirely of events in the original thread, but each event is interpreted as a plot element (which may involve attributing certain actions to the characters that were not explicit in the event, and attributing motivations to character behaviour). The second heuristics attempt to emulate to tell a story about some events in her day, intending to report faithfully the inspiring events but not necesarily the rest of her day. The person bases the story as faithfully as possible on the inspiring events, brings in some additional real events to support it, but may consider some fictional events to better match the story with a target plot. We refer to this approach as *imaginative enrichment*, because the material from the original thread can be enriched to adapt better to the plot of the story. In this approach, any additional events from the thread brought in to support the story need not be too faithfully rendered.

Overall, it seems that the two procedures proposed model different approaches to the task of generating stories about established facts. If the speaker wants to be careful in representing the events as they really happened, the imaginative interpretation procedure would be preferred. But the set of resulting stories may not include elaborate plots or fancy flights of fantasy. If the speaker is not so careful about representing the events as they happened, the imaginative enrichment might be preferred.

The *imaginative interpretation* heuristic involves applying the original procedure to the complete thread, but rejecting any allignments that do not include the events present in the target seed. This approach is a simple application of the original procedure with the additional constraint. It corresponds to ensuring that the part of the input thread that is being mapped onto a plot includes the desired events in the target seed.

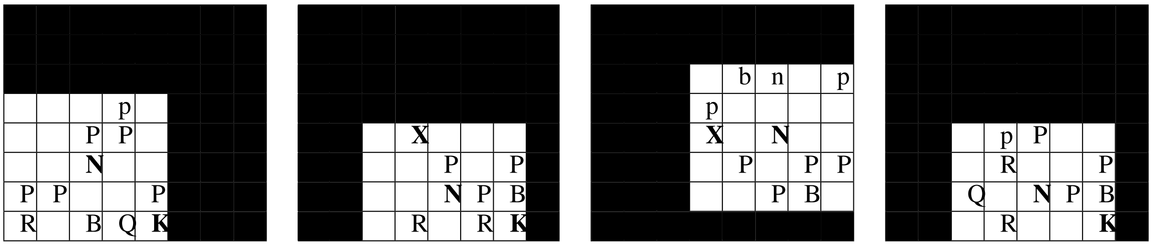
An example of a plot produced by a process of imaginative interpretation is shown in Table 6.

The *imaginative enrichment* heuristic goes one step further. The heuristic followed here involves two different improvements. First, the process of allignment is modified so that the matching of the events in the seed to plot elements in the plot can be optimised. This involves giving preference to allignments in which the targeted events match the plot elements assigned to them perfectly, even if the score of the complete allignment over the plot drops somewhat as a result. The process of allignment is therefore broken down into an initial allignment between the targed seed and the plot (which returns an allignment of the complete seed with a subspan of the plot), and a later stage of finding allignments for any remaining subspans of the plot with the subspans of the thread surrounding the target event. Second, in cases where part of the input thread – the events in the target seed and some additional support events – is mapped to partially instantiate a plot but some plot elements in the plot have not found a correspondence with real life events, the imaginative enrichment heuristic accepts the match as valid. Because of the nature of the process, the narrative roles in the unalligned plot elements will at that stage have been mapped to characters in the thread. The resulting plot is therefore coherent, even if some of the plot elements involved are not actually supported by events in the input thread.

Plots produced by imaginative enrichment will only differ from those produced by imaginative interpretation in that, in some case, no supporting events for some for their plot elements can be shown. The surface form of plots produced by either procedure is indistiguishable.

[1] The internal structure of a plot in terms of the interwoven plot spans from different axes of interest may play a significant role in the storyfication process once subplots start to be considered. For the present, it can safely be disregarded without affecting the outcomes.

Move: 7	Move: 37	Move: 39	Move: 55



character lwk (**N**) character wk (**K**) mutual_love lwk wk want_to_marry lwk wk lover lwk beloved wk	character rwk (**X**) (guardian rwk wk) opposed_to_plan rwk sundered lwk wk	(different_class lwk wk) high_status_revealed lwk ¬ sundered lwk wk	marry lwk wk

Table 6: Storyfication of the thread for the left white knight (lwk), targeted on events 37 and 55: the left white knight (lwk, represented in the diagrams as **N**) in terms of his romance with the white king (wk, represented in the diagrams as **K**) in the face of opposition of is guardian the right white knight (rwk, represented in the diagrams as **X** to distinguish him from the left white knight).

Selecting Adequate Input Parameters

The two proposed procedures have been tested using as inputs descriptions of chess games in algebraic notation. Because every game is different in terms of what the pieces do over the duration of the game, results obtained from a particular game have to be judged in the context of how suitable that game was for the production of stories based on the available set of plots. The procedures to be tested also require as input a choice of which piece to focalise on. Again, different choices of focaliser may influence the results, as some pieces may be more active than others, or have a chance to see more activity around them. Finally, the choice of which events are included in the target seed also affects the results, as, for each thread, certain events are more likely to lead to good stories than others. For these reasons, comparisons in terms of quantitative evaluation are only meaningful across different results for a given choice of inputs.

Nevertheless, the choice of input parameters to test should be informed. Table 7 reports on the overall results obtained by the non-targeted version of the system for a sample of 6 games, to show how this property fluctuates from game to game. Results include number of stories obtained overall for each game, and average number of stories obtained for each type of piece. The length of the game is included as it clearly affects the number of stories that can be obtained from it.

These results show that longer games tend to provide more stories, and that bishops, knights and rooks are likely to produce more stories than other pieces. The choices of game parameters and focalisers for the test reported in this paper are made accordingly.

In order to provide some indication of how suitable the particular context is for each test of the proposed procedures, the prior version of *StoryFire* is applied as to the same context in each case and the average score of the stories produced is reported.

Metrics for Targeted Storyfication

The two proposed heuristics differ in nature, so they have to be evaluated differently.

Imaginative interpretation finds solutions in which every plot element in the chosen plot is alligned with an event in the input thread. Results of this approach may be evaluated on the basis of the metric already defined for scoring results of storyfication (average percentage of set of roles involved in the plot elements actually assigned to the characters present in the events).

Imaginative enrichment allows the construction of stories in which events other than those in the target seed need not be supported by events in the input thread. Results of this approach may be evaluated on the degree to which the plot of the resulting story is supported by events in the input thread. This is represented as a simple ratio between the plot elements that are supported and the total number of elements in the plot.

In order to ascertain specifically whether a solution for targeted storyfication is sucessful, the following quantitative metrics have to be considered:

- average percentage of match in character assignment between thread events and plot elements that have been alligned to one another (%MTP for match between thread and plot)
- percentage of the available plots that it has been possible to succesfully instantiate by applying a given heuristic (%PSI for plots succesfully instantiated)
- percentage of the plot elements in the plot that have been alligned with events in the input thread (%DTS for degree of thread support)

Game	Stories	Moves in game	P	B	K	R	Q	K
1	139	40	7.5	11.0	12.0	9.5	5.0	5.0
2	167	58	7.9	15.0	10.5	10.0	6.5	6.5
3	213	104	9.4	17.0	15.5	14.5	7.5	7.5
4	226	120	10.3	17.0	15.5	17.5	10.0	10.0
5	209	90	9.5	17.0	17.0	16.0	9.5	9.5
6	119	32	6.5	10.0	9.0	6.5	5.0	5.0
Av.	178.8	74	8.5	14.5	13.3	12.3	7.3	7.0

Table 7: Fluctuation of number of valid stories (overall and per type of focaliser piece) obtainable from 6 different chess games for the set of available plots. Piece types are shown as: P for pawns, B for bishops, K for knights, R for rooks, Q for queens and K for kings.

Table 8 reports values for these metrics obtained for a number of possible configurations of the input parameters. Based on the discussion presented in the Section above on *Selecting Adequate Input Parameters*, the longest game in the test set was selected, and the thread for one of the knights. The set of possible target seeds is restricted to time points that are covered by a particular thread, because the thread does not include information for time points in which the focaliser sees nothing happening around its position. A number of possible target seeds has been picked at random, both for single and double events, and considering different types of groupings of the target events with respect to the overall length of the thread.

Discussion

The results for imaginative interpretation indicate a progressive decrease in the scores as more constraining targets are provided. Whereas the un-targeted storyfication reaches an average score on MTP of 97.9, the best score for a single event seed is 96.8 and the best score for a double event seed is 93.0. The relative position of the targeted events within the overall thread also seems to affect the scores. When a targeted event is close to the beginning or the end of the thread, this imposes a limit on the number of events that can be matched to parts of a plot before or after the target. As a result, the set of possible solutions is reduced. This has a drastical effect on the number of solutions produced, as indicated by the values for the %PSI metric for the double event target seeds as the targeted event approach the end of the thread.

The results for imaginative enrichment are more difficult to interpret. The procedure employed gives priority to focusing the plot on the event in the thread that best matches the character assignments in the plot, in contrast to the imaginative interpretation procedure that gives priority to a better score over the complete plot. This priority sometimes leads the results to plot with very poor support from the input thread.

The procedures described are at present simple baselines. Many computational aspects are in need of improvement. The chess domain is in itself also a very elementary case study. It is surprising how such a simple set up can yield insight on the mechanics of putting together the elements that go into a simple story, and how it allows consideration of issues relevant to the task such as the concept of targeting particular events during plot construction, or the difference between prioritising faithful reporting of observed events or construction of rich plots loosely based on some particular selection of the observed events.

The set of plots currently in use is also a first approximation and would also need to be expanded. The proposed representation in terms of axes of interest and plot schemas, articulated as plot spans, has proven to be a powerful tool for efficiently generating a variety of plots.

In its present form, the proposed approach to storyfication is based on the application of a set of pre-compiled plots to find the one that best matches an input set of events. The use of the words "story" and "plot" should not mislead the reader into thinking that the model described in this paper is intended as a plausible model of how humans address the task of giving birth to works of narrative of literary value. The processes involved in that nobler task are undoubtedly much more complex than the procedures outlined here. The use of "story" and "plot" is made to clarify the need for the desired outputs to satisfy basic restrictions in terms of being a particular type of sequential discourse, that includes a beginning, a complication and a resolution, concerns a character that can be clearly identified as a protagonist, and ends with a certain sense of closure. The ability that these procedures attempt to model is the simpler task of packaging a subset of the events one has observed over a given period, in such a way as to tell it in an entertaining manner to someone else. These simpler stories share with their literary counterparts some of their basic constraints but none of the complexity or the elaboration. In this sense, they constitute a good case study on which to break new ground over a simple representation.

Conclusions

The decision of basing a story on a particular set of events that will appear in its core has been shown to impose significant constraints on the task. Simple exploration of the alternatives available from a computational point of view indicates that an author faced with this task would have to choose whether to aim for faithful rendering of the context in which the selected events happened, or to give priority to the events themselves, and accept the possibility of building a new context for them that improve their potential as a story.

The proposed procedures and resources are intended as a first approximation to the task. Several avenues for future research have been uncovered. More elaborate refinement

	Seed	ImagInt		ImagEnr	
		%MTP	%PSI	%DTS	%PSI
No seed	-	97.9	100		100
Single event seed	(7)	95.9	100	98.3	100
	(37)	96.8	100	27.0	100
	(72)	69.4	100	44.4	100
Double event seed	(7, 14)	93.0	100	85.5	100
	(37, 46)	85.4	90	100.0	100
	(55, 72)	80.5	60	97.7	100
	(7, 72)	73.0	20	75.3	100

Table 8: Results of storyfication: for the left white knight, on game 4 involving 120 moves, showing different choices for the target seed and their impact on the metrics: MTP for match in character assignment between thread events and plot elements, PSI for percentage of plots succesfully instantiated, DTS for degree in which the plot elements in the plot is supported by events in the thread. In each case, averages over the complete set of stories produced for the given input are given. All scores are normalised to 100 for ease of comparison.

of the procedures from a computational point of view will be addressed. The set of plots considered will be expanded. Applications beyond the chess domain case study will be considered.

Acknowledgments

This paper has been partially funded by the project IDiLyCo: Digital Inclusion, Language and Communication, Grant. No. TIN2015-66655-R (MINECO/FEDER) and the FEI-EU-17-23 project InViTAR-IA: Infraestructuras para la Visibilización, Integración y Transferencia de Aplicaciones y Resultados de Inteligencia Artificial.

References

Abbott, H. 2008. *The Cambridge Introduction to Narrative*. Cambridge Introductions to Literature. Cambridge University Press.

Booker, C. 2004. *The Seven Basic Plots: Why We Tell Stories*. The Seven Basic Plots: Why We Tell Stories. Continuum.

Bruner, J. 1991. The narrative construction of reality. *Critical inquiry* 1–21.

Flower, L., and Hayes, J. 1981. A cognitive process theory of writing. *College Composition and Communication* 32(4):365–387.

Genette, G. 1980. *Narrative discourse : an essay in method*. Cornell University Press.

Gervás, P.; León, C.; and Méndez, G. 2015. Schemas for narrative generation mined from existing descriptions of plot. In *Computational Models of Narrative*. Atlanta, Georgia, USA: Scholoss Dagstuhl OpenAccess Series in Informatics (OASIcs).

Gervás, P. 2009. Computational approaches to storytelling and creativity. *AI Magazine* 30(3):49–62.

Gervás, P. 2012. From the fleece of fact to narrative yarns: a computational model of composition. In *Workshop on Computational Models of Narrative, 2012 Language Resources and Evaluation Conference (LREC'2012)*.

Gervás, P. 2013. Stories from games: Content and focalization selection in narrative composition. In *I Spanish Symposium on Entertainment Computing*.

Gervás, P. 2014. Composing narrative discourse for stories of many characters: a case study over a chess game. *Literary and Linguistic Computing* 29(4).

Gervás, P. 2018. Storyfying Observed Events: Could I Dress This Up as a Story? In *AISB Symposium on Computational Creativity (submitted)*.

Herman, D. 2003. *Narrative Theory and the Cognitive Sciences*. CSLI Publications. CSLI Publications.

Herman, D. 2004. *Story Logic: Problems and Possibilities of Narrative*. Frontiers of narrative. University of Nebraska Press.

Herman, D. 2013. *Storytelling and the Sciences of Mind*. Cambridge, MA.

Kybartas, B., and Bidarra, R. 2016. A survey on story generation techniques for authoring computational narratives. *IEEE Transactions on Computational Intelligence and Artificial Intelligence in Games*.

Mueller, E. T. 2003. Story understanding through multi-representation model construction. In *Proc. of the HLT-NAACL 2003 Workshop on Text Meaning - Volume 9*, 46–53. Stroudsburg, PA, USA: ACL.

Nisbett, R. E., and Wilson, T. 1977. Telling More Than We Can Know: Verbal Reports on Mental Processes. *Psychological Review* 84(3):231–259.

Propp, V. 1928. *Morphology of the Folk Tale*. Akademija, Leningrad.

Schank, R., and Abelson, R. 1977. *Scripts, Plans, Goals and Understanding: an Inquiry into Human Knowledge Structures*. Hillsdale, NJ: L. Erlbaum.

Winer, D., and Young, R. M. 2016. Discourse-driven narrative generation with bipartite planning. In *Proceedings of the 9th International Natural Language Generation conference*, 11–20. Association for Computational Linguistics.

Deep Learning-based Poetry Generation Given Visual Input

Paper type: Technical Paper

Malte Loller-Andersen and Björn Gambäck
Department of Computer Science
Norwegian University of Science and Technology
7491 Trondheim, Norway
malte.loller-andersen@knowit.no gamback@ntnu.no

Abstract

The paper describes the implementation and evaluation of a system able to generate poetry satisfying rhythmical and rhyming constraints from an input image. The poetry generation system consists of a Convolutional Neural Network for image object classification, a module for finding related words and rhyme words, and a Long Short-Term Memory (LSTM) Neural Network trained on a song lyrics data set compiled specifically for this work. In total, 153 stanzas were generated and evaluated in two different experiments. The results indicate that the deep learning based system is capable of generating subjectively poetic, grammatically correct and meaningful poetry, but not on a consistent basis.

1 Introduction

Computational linguistic creativity involves theoretical study of language as well as developing computer algorithms to process and generate language. It uses a varied arsenal of machine learning, artificial intelligence and statistics to generate, predict and extract the meaning of texts, such as story narratives, jokes, analogies, word associations, and poetry. For language generation, poetry is one of the more interesting and complex challenges, since its value depends on both form and content. In this paper, a state-of-the-art poetry generator is designed and implemented. It takes an image as input and uses Inception (Szegedy et al. 2016), a pre-trained convolutional neural network (CNN) image classifier, to find objects in the image. The poetry generator then returns a short stanza based on the objects, by combining tree search with a Long-Short Term Memory (LSTM) recurrent neural network (RNN) trained on a custom made data set built from scratch from $200,000+$ songs. Instead of using rule-based and fill-in methods, the system actively predicts outcomes in a creative way. Strophically correct poems are guaranteed by combining tree search with deep learning, searching for optimal paths with suitable rhyme words.

Poetry has various different forms and is a very subjective genre, therefore a concrete definition of poetry can be hard to nail down. The definition used in this paper is:

Poem. *A set of lines satisfying rhyming and rhythmic constraints. The text generated consists of one stanza containing Y lines. The lines have to be a length of X syllables, and a line must rhyme with another line.*

By using this definition, it is clear what kind of poetry is being generated, and conclusions can be based on this. Furthermore, a stanza is a group of lines separated from others in a poem, often to shift between action, moods and thoughts. The terms stanza and poetry will be used interchangeably about the poetry generated by this system.

The next section describes the state-of-the-art in poetry generation, focusing on corpus-based methods and deep learning approaches. The data set gathering and the preprocessing of the songs are outlined in Section 3. Section 4 talks about the design and implementation of the poetry generation system. Section 5 shows the results gathered from the system, which are then discussed and evaluated in Section 6, showing that participants in a survey were able to identify a system generated poem vs a human generated poem with a 76.5% success ratio, with the system being able to generate stanzas that were perceived as aesthetically good, but not by everyone and not consistently. Finally, the conclusions are drawn and possible future work is discussed.

2 Related Work

Deep learning has proven very successful in image recognition where convolutional neural networks have heavily dominated in recent years. Most of latest efforts in poetry generation also use deep learning, although the research field itself started developing already in the 1990s, with multiple different methods being tried out. The high complexity of creative language creates substantial challenges for poetry generation, but even though the task is complex, many interesting systems have been developed. Gervás (2002) roughly divided poetry generation into four types of approaches: *template-based* (e.g., PoeTryMe by Oliveira, 2012 and Netzer et al.'s 2009 Haiku generator), *generate and test* (e.g., WASP by Gervás, 2000 and Tra-La-Lyrics by Oliveira, Cardoso, and Pereira, 2007) , *evolutionary* (e.g., POEVOLVE by Levy, 2001 and McGonagall by Manurung, 2004), and *Case-Based Reasoning* approaches (e.g., ASPERA by Gervás, 2001 and COLIBRI by Díaz-Agudo, Gervás, and González-Calero, 2002). Oliveira (2017) updates and extends Gervás' classification, while Lamb, Brown, and Clarke (2017) introduce a slightly different taxonomy. However, the focus in recent years can really be said to have shifted to two types of approaches, corpus-based and deep learning-based methods.

Corpus-based methods aim to find other poems and use their structure to create new poems. They often use multiple corpora and substitute words based on their part-of-speech (POS) tags and relevance. Colton, Goodwin, and Veale (2012) presented Full-FACE Poetry Generation, a corpus-based system which uses templates to construct poems according to constraints on rhyme, metre, stress, sentiment, word frequency, and word similarity. The system creates an aesthetic, a mood of the day, by analyzing newspapers articles, and then searches for an instantiation of a template maximizing the aesthetic.

Toivanen et al. (2012) introduced a system using two corpora, a grammar corpus and a poetry corpus, in order to provide semantic content for new poems and to generate a specific grammatical and poetic structure. The system starts by choosing a topic, specified by a single word. Topic associated words are then extracted from a background graph, a network of associations between words based on term co-occurrence. A desired length text is then randomly selected from the grammar corpus, analyzed and POS-tagged, and each word is substituted by words associated to the topic. Toivonen et al. (2013) extend this model and show how simple methods can build surprisingly good poetry.

Zhang and Lapata (2014) made one of the earliest attempts at generating poetry using *deep learning*. The poem is composed by user interaction, with the user providing different keywords, that has to be words appearing in the *ShiXueHanYing* poetic phrase taxonomy. The generator creates the first line of the poem based on the keywords and then expands the keywords into a set of related phrases that are ranked using multiple character-level neural networks. The system is very complicated and computationally heavy, using a CNN and two RNNs, but yields respectable results.

Wang, Luo, and Wang (2016) proposed an architecture using an attention-based recurrent neural network, which accepts a set of keywords as the theme and generates poems by looking at each keyword during the generation. The input sequence is converted by a bi-directional GRU (gated recurrent unit) encoder to a sequence of hidden states. These hidden states are then used to regulate a decoder that generates the poem character by character. At each time step, the prediction for the next character is based on the current status of the decoder and all the hidden states of the encoder.

Wang et al. (2016) used a planning-based recurrent neural network, inspired by the observation that a human poet often makes an outline before writing a poem. The system takes a user's input which can be either a word, a sentence or a whole document, and generates the poem in two stages: First, in the *poem planning stage* the input query is transformed into as many keywords as there are lines in the poem, using TextRank (Mihalcea and Tarau 2004) to evaluate the importance of words. However, if the user's input query is too short, keyword expansion is done using both an RNN language model and a knowledge-based method, where the latter aims to cover words on topics that are not in the training data. Then in the *poem generation stage*, the system takes all previous generated text and the keyword belonging to a given line as input, and generates the poem sequentially line by line. The generator uses the same encoder-decoder structure with GRUs as in (Wang, Luo, and Wang 2016), but slightly modified to support multiple sequences as input.

Ghazvininejad et al. (2016) proposed Hafez, a system that creates iambic pentameter poetry given a user-supplied topic. It starts by selecting a large vocabulary, and computes stress patterns for each word based on CMUdict,[1] an open-source machine-readable pronunciation dictionary for North American English that contains over $120,000$ words. A large set of words related to the user topic are retrieved using the continuous-bag-of-words model of *word2vec* (Mikolov et al. 2013). Next, rhyme words are found and put at the end of each line. Also using CMUdict, the system tries to find rhyming words with related words. However, as a fall-back for rare topics, fixed pairs of often used words are added. A Finite-state-acceptor (FSA) is built, with a path for every conceivable sequence of vocabulary words that obeys formal rhythm constraints. A path through the FSA is selected using a RNN for scoring the final outcome. The RNN uses a two layer recurrent neural network with long short-term memory, trained on a corpus of $94,882$ English songs.

Most similar to the present work is the recent effort by Xu et al. (2018) to use an encoder-decoder model to generate Chinese poetry. They utilised the poem data set of Zhang and Lapata (2014) together with images collected from the Internet that depicted key words contained in the poems. In the encoder part, a CNN extracts visual features from the input images, while semantic features from previously generated lines of the poem are built by a bi-directional Gated Recurrent Unit (GRU). The decoder then consists of another GRU that generates a new line of the poem (Chinese) character by character, using the visual and semantic features together with the keywords.

3 Data set

Optimally, a data set should consist of data as close as possible to the desired results of the system, since the goal of training computer models is to mimic the data set as closely as possible. However, collecting a data set consisting of poetry is very inconsistent, meaning that every poet writes differently and uses different words and expressions. The result is that the available sample of reasonably consistent poetry is rather small. In comparison, hundreds of thousands of song lyrics — essentially rhythmic poems — are readily available. Hence, those in the field of poetry generation commonly use song lyrics rather than poetry as their data sets. Therefore, just like Ghazvininejad et al. (2016), song lyrics were chosen as a base for the data gathered for this project, since other data sets of the right size and content are not available due to various copyright protections. For the same reasons, the collected data set cannot be distributed further.

The data set was collected from `www.mldb.org`, an online song lyrics database. A Python script was written to connect to their site, sort through the HTML files of the site and find the song text, artist and album. Beautiful Soup[2] handles the HTML file by creating a parse tree, making it easy to navigate and handle the data provided in the HTML

[1] `www.speech.cs.cmu.edu/cgi-bin/cmudict/`

[2] `www.crummy.com/software/BeautifulSoup/`

file. A total of $206,150$ songs were collected, with a vocabulary of $391,363$ unique words and $46,346,930$ tokens in total. This dataset was filtered to remove songs that contained less than 95% of English words, as well as non-lyric content such as escape characters (e.g., newline, backslash and quote characters), custom messages by the sender (e.g., "Submitted by x" and "Thanks to y"), name of artists, verse and chorus notation, and notation of physical movements. After filtering, $80,608$ songs remained with a vocabulary of $91,097$ unique tokens. This is still too large and sparse a dataset for efficiently training a classifier, so all songs containing words that are not in the top $30,000$ vocabulary were also removed, leaving $40,685$ songs ($8,712,213$ tokens) with a final vocabulary size of $27,601$.

4 System Design and Implementation

The system consists of several stages. First, after an image is obtained it is run through Inception for classification. The output from Inception is the five top scoring classes. If these are scored lower than a set threshold, the result is discarded; otherwise they are used to find related words using ConceptNet by utilising ConceptNet's related words feature to return a scored list of related concepts, and then using the core of ConceptNet to find edges for the concepts, with the concept at the other end of the edge being saved and scored. When all related words have been found, rhyme words are explored, with the aim that both rhyming words should be related to the picture.

The next stage is to find the optimal path through a tree structure to construct a sentence. The path starts from the start of the sentence and ends on the rhyme word at the given line. Any number of lines can be generated. When a line is generated, an attempt to check the grammatical structure is made, but its use is limited. Figure 1 shows all steps from an input image to the generated poetry.

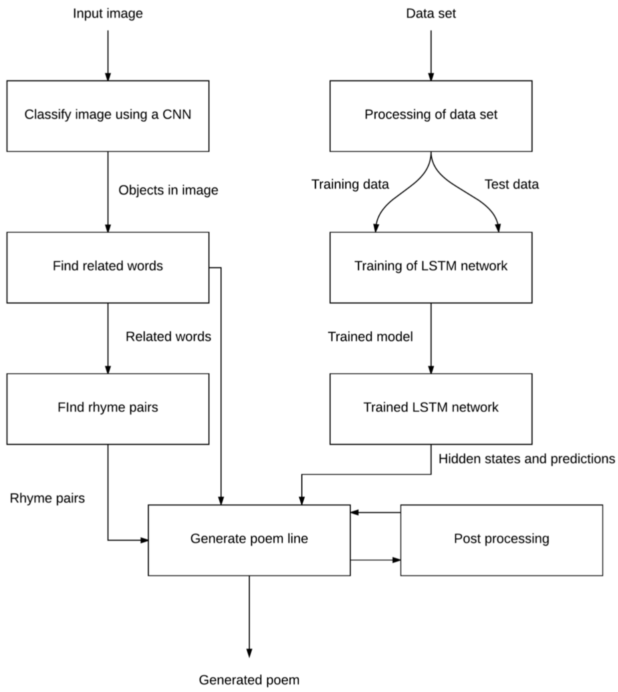


Figure 1: A high level overview of the system

Finding rhyming words from images

Looking at the components of the architecture in more detail, Inception-v3 (Szegedy et al. 2016) is the **object recognition** part, so the images used in the experiments must contain at least one object recognizable by Inception-v3, which is a 42 layer convolutional neural network available through TensorFlow in both Python and C++. Inception-v3 is pretrained on the 2012 ImageNet Large Visual Recognition Challenge (ILVRC), where the task is to classify a picture into one of 1,000 classes. The classes have no particular themes, however, 400 of them are animals, while the rest are quite diverse and can be anything from *CD player* to *castle*.

The keywords returned by the CNN are used to gather a large set of **related words**, based on the related words feature in ConceptNet (Speer and Havasi 2012), which returns a number of words along with a similarity score between 0 and 1. The next step looks at the edges of the concepts belonging to the keywords: each edge has another concept end point, and these are retrieved along with a score, which is between 1 and 12, but normalized to $[0, 1]$. The scores from related words and nearby concepts are treated equally. If the system has few high scoring related concepts, the system will prune the highest scores and retrieve concepts related to them. This repeats until the system has at least 200 related words. All related words need to be in the vocabulary. The more related words that are gathered, the more options the system has to chose from. At 200, or more, words the system seemingly has no problems finding applicable related words. The risk of finding no appropriate related words, or using poorly scored related words is that the poem will not be perceived as relevant to the image given as input.

When the related words are found, the system looks for **rhyme pairs**. This is done using CMUdict, which finds the pronunciation of a given word. For instance, the word *dictionary* is returned as *D IH1 K SH AH0 N EH2 R IY0*, where the numbers are the syllable stress and the letters the pronunciation. Two rhyming rules are used to find the rhyming words. The first checks if the endings of the words are pronounced the same. The length of the rhyme does not matter, though it must be at least one syllable long. The second rule states that the consonant sounds preceding the rhyme must be different. A consequence of this is that slant rhymes are not allowed in the system. Therefore, mostly perfect rhymes are present in the rhymes. This is a design choice, because slant rhymes such as *milk—talk* do not sound particularly good. However, if the system cannot find any suitable rhyming words, the second rhyming rule is discarded, and only the word endings are checked. Ideally, both rhyme words should be related to the initial image, but if there are no related words rhyming with each other, the system looks for other words rhyming on a related word in the vocabulary. The highest scoring related words and their rhyming words are preferred.

Predicting word sequences

The Long Short-Term Memory (LSTM) network then has the task of predicting the next word in a given sequence. It takes a word and the previous hidden state as an input, before producing an array of the scores for each other word in the vocabulary. The hidden state is then updated taking the whole sequence of previous words into consideration. The vocabulary is embedded into a dense Vector Space Model representation before being fed further into the network. The embedding matrix is initialized randomly, but as the training goes on, the model learns to differentiate between words by looking at the data set. To save memory when training, batches of the data set are converted into tensors that TensorFlow can use to train the model.

When training a network, several different parameters have to be set. Following Zaremba, Sutskever, and Vinyals (2014), the most important variables experimented with were: batch size, learning rate and learning rate decay, probability of keeping a neuron (dropout; Srivastava et al., 2014), number of steps unrolled in the LSTM layers, and parameter initialisation. The learning rate starts at 1.0, but after 14 epochs it is decreased by a factor of 1.15 after each epoch. The batch size is 20, and the parameters are uniformly initialized in the range $[-0.04, 0.04]$. The dropout rate is 65% for non-recurrent units. The number of steps unrolled in the LSTM layers is 35. The training time for the best performing network was 34 hours and 41 minutes on a NVIDIA GeForce GTX 970 with 4 GB memory.

Several different architectures of the LSTM network were tried, with the model with the lowest word perplexity score during training, and therefore best performing, having four layers: one input layer to represent the words coming in from the data set, followed by two hidden layers with the size of 1100 cells, and one softmax layer to give the predictions for the next words. The core of the network are the LSTM cells found in the two hidden layers that compute the possible values for the next predictions.

The loss function the network tries to minimize is the average negative log probability of the target word: $-\frac{1}{N}\sum_{i=1}^{N}\ln(p_{\text{target}_i})$ where N is the total training set size, *target* is the target word, and i is the word being looked at. The log loss is the cross entropy between the distribution of the true labels and the predictions given by the network. Therefore, by minimizing the log loss, the weights are optimized to produce a precise distribution of true labels. This is equivalent to maximizing the product of the predicted probability under the model. Per-word perplexity then measures how well the LSTM network performs during training:

$$e^{-\frac{1}{N}\sum_{i=1}^{N}\ln(p_{\text{target}_i})} = e^{loss} \tag{1}$$

A gradient descent optimizer is used along with backpropagation to minimize the loss function during training. The backpropagation algorithm uses the error values found by the loss function, *L*, to calculate the gradient of the loss function with respect to the weights, *w*, in the network, $\frac{\partial L}{\partial w}$.

To optimise the architecture with respect to word perplexity, the number of layers, hidden units, and epochs were varied. Zaremba, Sutskever, and Vinyals (2014) achieved a perplexity of 68.7 on the Penn Tree Bank data set, with a vocabulary of only $10,000$ words. Usually when training on larger vocabularies, the network should be more confused, so perplexity should increase. Here the vocabulary size is $27,601$, but the tested networks still achieved remarkably low perplexity: with only one hidden layer and 1100 hidden units, the LSTM's perplexity after 53 epochs was 65.0. With two hidden layers and 1100 units, perplexity dropped further, to 36.7 after 53 epochs. The reason for the low perplexity is the data set: song lyrics tend to use a small variety of words and often repeat lines. The network tries to mimic this behaviour. The result is that the predictions of rare words are often 0.0, with major implications for system performance.

Generating poetry

The poetry generation takes the related words, rhyme pairs and a trained LSTM network as input. It creates a tree structure to find the highest scoring paths through the tree, applies part-of-speech tagging to the paths, and returns the most optimal path that fits the grammar constraints. The stanzas are chosen to be 4 lines long, with each line being 8 syllables long, and the rhyming scheme being *AABB*. One stanza is generated per image. First, a tree is generated for each line in the stanza. One node symbolizes one word and its syllables, and each edge is the score between one node and another, as provided by the trained LSTM network. The root of the tree is an empty string with the initialized hidden state. Based on the root and the hidden state, the system takes the top 30 words predicted and sets them as the root's children. For these, the system generates 30 predictions. This process is done when the syllable constraints are fulfilled.

However, this tree is too large to be effective and the search takes too long. In the worst case scenario, when all words are monosyllabic, a tree finding the path to an eight syllable length phrase will consist of 30^8 nodes. Due to various calculations in TensorFlow, only about five nodes can be looked at per second, so searching through that many nodes is unacceptable. The solution to this is two-folded: First, a depth first search is performed while deleting all child nodes that have already been looked at. Second, the tree is pruned based on the score of a line or syllabic constraints.

The depth first search takes the syllable constraints into account. When the search reaches a leaf node, the system generates a score of how good the line is, which is the sum of each prediction for each word. The line can be positioned anywhere in the stanza, depending on the rhyming scheme chosen for the stanza. A tuple is generated that contains the rhyme words for two different lines, and scored based on how relevant the words are to the image. The tree search then goes back to look at the leaf node's parent, sets the current node to be looked at as one of the siblings of this node, and calculates the next predictions for this new node. This is done 30 times, once for each of the 30 top predictions.

Pruning of the tree is done in two ways: Based on the score, or based on different syllabic constraints. A node is skipped if does not fulfil the following constraint:

$$node_{\text{score}} > \frac{S_{\text{node}}}{S_{\text{tot}}} \times \frac{top\_score[-1]}{2} \tag{2}$$

where S_{node} is the syllable count for a given node, S_{tot} the total number of syllables needed to finish the line, and $top\_score[-1]$ the worst top score achieved at this point.

The score of a given line is calculated in two parts, with one coming from the words generated in the sentence and the other being the prediction of the rhyming word. These values are continuously normalised against other values in the top score list, so that the line score and the rhyme word prediction are evaluated as equal parts — otherwise, the line score would dominate, since it is the sum of multiple predictions, while the rhyme word prediction is only one.

The system tends to repeat words, since the network is trained on song lyrics that often are repetitive. Therefore words already used in a line receive lower scores, and are avoided if possible. Furthermore, the system prefers to use easy and safe words. Because of this, a higher score is given to the related words. This forces the system to explore new words and hence enhances performance. However, completely avoiding repetition of words also has a disadvantage, since repetition is an important poetic feature.

A problem the system cannot handle very well is the transition between the generated line and the rhyme word. POS-tagging is added to address this problem. The tagger first tags the line generated, and then the rhyme word separately. However, tagging a word without any context is not possible, so the 1M word POS-tagged Brown corpus is used to determine the most frequent tag of the rhyme word. By fixing the rhyme word at the end of the line, and POS-tagging again, this time the line and the rhyme word together, the rhyme word gets another POS-tag. If this tag matches the most frequent one in the Brown corpus, the line is accepted. Clearly, it is not always enough to check that the POS-tag matches the most frequent tag; however, it guides the system to generate grammatically better poems.

5 Experiments and Results

Two different experiments were conducted to test the system. In the first, human subjects selected any image of their choosing, preferably one they had taken themselves. This image was then run through the system, and the subject evaluated the generated poems according to three criteria described below. To avoid testing the system on images and to ensure using the best generated poetry, the participants were first asked to find at least three images of their choosing. Each image was then run through the system and the stanzas retrieved, which the participants were asked to rate.

In the second experiment, the subjects were tested to see if they could differentiate between poetry generated by the system and poetry written by a human. Here, a participant was shown an image and the corresponding human generated and machine generated stanzas, and asked which stanza was the human generated one.

Finally, the participants were asked about their overall thoughts of the system. 46 persons participated in the experiments, with no requirement that they should have any prior experience of writing or evaluating poetry. A total of 153 stanzas were rated on the three criteria, and a total of 38 evaluations were done to decide if the poetry was from a human or a computer.

The sun is in my big raincoats
I don't know what to do scapegoats
I'm raining and it looks like rain
There's so much for me to abstain

Score: [3.0, 2.7, 2.0]

Figure 2: "Red Umbrella" by DLG Images (CC BY 2.0) www.flickr.com/photos/131260238C%40N08/16722739971/.

I don't know why it feels like crabs
That make me want to look at cabs
So come on lets get out of zoo
And dive into my big canoe

Score: [3.0, 2.5, 1.0]

Figure 3: "Jellyfish" by Jennifer C (CC BY 2.0) www.flickr.com/photos/29638108%40N06/34440131755/.

I want to be with your tent group
In the shape of your hand and troop
My life is an operation
So come on lets go now station

Score: [2.5, 3.0, 2.5]

Figure 4: "KFOR 12 training", The U.S. Army (CC BY 2.0) www.flickr.com/photos/soldiersmediacenter/3953834518/.

Grammaticality, poeticness and meaningfulness

State-of-art poetry generation commonly use either automatic or human-based evaluation methods. The automatic approaches, such as BLEU (Papineni et al. 2002), originate in machine translation research and assume that human written references are available, which is not the case for the present work. Another problem with these evaluation methods is that they have been found to have little coherence with human evaluation (Liu et al. 2016).

Hence we will here focus on human-based evaluation, in particular along the lines of Manurung's (2004) criteria *grammaticality* (lexical and syntactic coherence; in essence ruling out random sequences of words), *meaningfulness* (convey a conceptual message which is meaningful under some interpretation; here it will also include topic coherence with the visual input), and *poeticness* (phonetic features such as rhymes and rhythmic patterns). Each of the dimensions is scored on a scale of 1–3. A slightly different version of these criteria has also been proposed: *fluency*, *meaning*, *poeticness* and *coherence* (Yan et al. 2013; He, Zhou, and Jiang 2012). However, no in-depth explanation of these criteria has been made and different definitions of the four metrics have been given, so this work will use the metrics of Manurung (2004) and the goal of the first experiment was to evaluate each poem on those criteria. The average score and standard deviation of all poems evaluated were 2.614 ± 0.519 for poeticness, 2.381 ± 0.470 for grammaticality, and 2.050 ± 0.712 for meaningfulness, with median scores of 2.8, 2.5, and 2.0, respectively.

Hole in the back of my bottle
I don’t know what it’s like throttle
It makes me feel this way grape wine
And when I’m with you flask align

Score: [2.8, 2.5, 2.5]

Figure 5: “NAPA 2012” by cdorobek (CC BY 2.0) www.flickr.com/photos/cdorobek/8093546797/.

I don’t know what to do cute bear
The way you look at me when fair
And I’m so in love with teddy
You’re just a part of already

Score: [2.8, 2.7, 3.0]

Figure 6: Picture by Yumeng Sun, used with permission

Five randomly selected poems are shown in Figures 2–6 to display a variety of different scored poems. The input image is to the left while the generated poem related to the image is on the right, followed by the score for the poem, in the order of poeticness, grammaticality, meaningfulness.[3] As can be seen, poeticness scores higher than grammaticality and meaningfulness. The reason for this is the guarantee of the lines being eight syllables long, and almost every line rhymes. When the system cannot find suitable rhyme words, sub-optimal rhyme words are chosen.The grammaticality score is a bit lower, and the meaningfulness is slightly above the “partially meaningful” level. Both of these are influenced by the system predicting line ending rhyme words with a 0 probability. This happens when the LSTM network dislikes all the rhyming words found, scoring them all as 0.

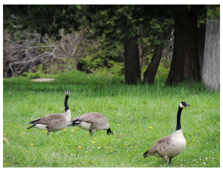
It is also interesting to compare the results of poems with two or more lines where the rhyme word is predicted by a non-zero number against poems containing zero or one such lines, since when the rhyme word is predicted by a zero value, the POS-tagging and the word relevance decide the rhyme word and the corresponding line. There can be two reasons for a zero score: that the network has been trained on the word but does not consider it a good fit, or that the training data contained too few occurrences of the rhyming word so that the network is uncertain about its fitness. 33 of the 153 stanzas contain two or more lines where the rhyme word was predicted by a non-zero value. Looking at the scores for those only, it is clear that when it is known that the system has an opinion of the rhyming word, performance increases (note that the opinion not necessarily has to be good). The average poeticness goes from 2.614 to 2.793, grammaticality increases from 2.381 to 2.691, and poeticness from 2.050 to 2.464. However, the Pearson Correlation Coefficient reveals that even though there technically is a positive correlation between the number of non-zero predictions for the rhyme word and survey scores, the relationship

[3]Unless stated otherwise, images are used under Creative Commons Attribution 2.0 license (CC BY 2.0): `creativecommons.org/licenses/by/2.0/legalcode`

It’s been a long time and I’ve bees
But I don’t want to get disease
I’m all out of love, oh so sweet
Just give me one more chance to meet

I am flying in the warm air
The other bees are very fair
It is a beautiful sunday
I am leaving later today

Figure 7: “bee” by David Elliott (CC BY 2.0) www.flickr.com/photos/drelliott0net/15105557167/



I want to see swans in the sky
You and me, we are in for fly
If there’s a lake out there for goose
There is no limit for excuse

Eating dinner like a wild goose
Food is something that I can’t lose
Although cooking is really tough
Nothing better can make me bluff

Figure 8: “Canada Geese” by Kevin M. Klerks (CC BY 2.0) www.flickr.com/photos/ledicarus/34618458382/.

is weak: the correlation score for poeticness was 0.1658, 0.3040 for grammaticality, and 0.2674 for meaningfulness.

Human- and machine generated poetry experiment

The second experiment was done to see if the participants could differentiate between the machine generated poetry and human written poetry. The process was as follows: After 30 stanzas had been evaluated, the four highest rated poems were selected. The persons who had chosen those images were asked to write a poem with the same syllabic and rhyming constraints, i.e., each sentence had to have eight syllables, and the two first and two last lines should rhyme.

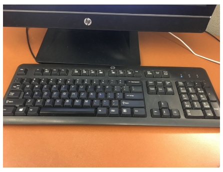
For all these four images, the computer generated stanza and human evaluated stanza are shown in Figures 7–10. The top stanza next to each image is the computer generated one, and the stanza in italics is the human written. The question asked to the participants of the experiment was: “Which stanza is human generated?” 38 persons evaluated the four items, giving a total of 152 evaluations. The participants chose the correct option in total 117 times, while the wrong option was chosen 35 times, so the participants had a 76.5% success ratio. Figure 8 was the easiest for the participants to single out, with only 5 of 38 participants answering wrongly. For Figures 7, 9, and 10, respectively 10, 11 and 9 participants erroneously picked the machine generated stanza.

The last part of the experiment was to get the participant sentiments. The question asked to the participants was: “After seeing the results of the images that you chose, and the poetry of the four images of experiment two, do you have any final thoughts?” 29 of the 46 participants used the word “fun” when evaluating the poetry, while 20 participants used the word “random”, and a few used the word “bad”. Other notions such as “interesting system” and “the poems feel alive” were also common. Seven participants mentioned “personification” as a *core value* of the generation system.

Broomstick on the back of my broom
Move on like dead man can assume
I don't know what to do and sweep
It makes me think that I'm asleep

Checking the status of my broom
I need the broom to sweep the room
I have no choice to keep waiting
It's hard for me to stop hating

Figure 9: "broom" by danjo paluska (CC BY 2.0) www.flickr.com/photos/sixmilliondollardan/3074916976/.



It's been a long time since the kick
It makes me want to hold you stick
And I know communication
I've got so much combination

Tapping on my keyboard at night
Hoping it will not cause a fight
You are my only listener
Don't let me be your prisoner

Figure 10: Picture by Yumeng Sun, used with permission

6 Discussion

The average score of 2.614 in the 'poeticness' category tells that people overall found the stanzas in between *partially poetic* and *poetic* in terms of rhythm and rhyming. However, the standard deviation is quite high (0.519), either representing a big spread of the sentiment of the participants, or a big spread in the quality of the poems generated. The average score of 2.381 for 'grammaticality' indicates that the language is closer to being *partially grammatically correct* than *grammatically correct*. The standard deviation of 0.470 indicates that the spread is smaller than for poeticness, but it is still quite large. The average score of 2.050 for the 'meaningfulness' category tells that the system is on average evaluated as *partially meaningful*, although the standard deviation of 0.712 indicates that meaningfulness is the most inconsistent property of the poems generated by the system.

Several effects of using song lyrics as training data for the LSTM network are reflected in the poetry, including: extensive use of the first person pronoun (*I*) as well as of the word *love* and terms related to it, repetition of some phrases (e.g., "I don't know", "I want to", "It's been a long time since", "So come on"), and that a large part of the vocabulary is predicted with zero probability at any given point in time. As a result, and as can be seen in the examples above, a large fraction of the generated stanzas include at least one form of *I*: 87% had at least one first person pronoun, 42% had two, and 12% of the poems had three. The extensive use of pronouns can lead to personification (non-human objects taking on human characteristics), with the poems often gaining human traits even though there are no humans in the image. Furthermore, *love* appears 10,671 times in the dataset and is found in 7% of the songs, leading to it appearing in 11% of the poems generated, which makes the poetry love based.

The system has a couple of limitations. The first one is Inception: if it misclassifies the input image, the system will perform poorly, because all related words and rhyme words will become related to the wrong class. Another limitation is that only the top-1 class is used to find related words. The reason to only use the top-1 result is that other results are often not related and might be wrong, therefore hurting the performance of the system. This means that only one object in the image is used. Using object recognition instead of classification might yield better results, since this would make it possible to identify multiple objects in the image. This could make the generated poems more dynamic as the poem can choose from a broader set of related words, and mention multiple objects in the image.

7 Conclusion and future work

A system was designed and implemented to take an image as input and generate poetry related to the image. The system includes a CNN for object classification, a module to find related words and rhyme pairs, and an LSTM network to score a tree search where nodes are representing the words being generated. 153 stanzas were generated and experiments were conducted with volunteering participants to evaluate the quality. The results of the system were varied, with a big standard deviation on the three criteria it was evaluated on. The system was not able to consistently generate stanzas that are perceived by everyone to be aesthetically good. The best results were achieved when the LSTM network could predict the rhyme word with non-zero prediction scores, however, only a weak correlation was found between the evaluation and the stanzas containing non-zero predictions.

A data set of more than 200,000 songs was gathered and pre-processed to train the LSTM model. Various difficulties arose when gathering the data, the biggest being the variety of random content in the song lyrics. Upon closer inspection of the data set some content that is not song lyrics are still present, however, the fraction of this content does not have any noticeable impact on the system. Using song lyrics is interesting due to elements of personification emerging and the grammar in a line is usually good until the rhyme word appears, while the system has trouble predicting rhyme words. Overall the system will probably never write any poetic masterpiece, but the evaluations made by the participants indicate that some generated stanzas were subjectively enjoyable. This suggests that the implementation could be useful as a foundation for other poetry generation systems.

One option for trying to enhance the performance of the predictions is changing the LSTM network to a Sequence to Sequence Model (Cho et al. 2014). Several newer poetry generation systems such as Wang et al. (2016) use this approach, and report good results. Another possible change to the system is to train a separate model for fetching related words. One popular model for doing this is to train a word2vec model. Implementing this into the system could enhance performance by finding better related words and better rhyming words. Training the LSTM network on poetry instead of song lyrics is also an interesting variation to test. Poetry has different properties than song lyrics, such as using a bigger variety of words more often.

References

Cho, K.; Van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; and Bengio, Y. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Colton, S.; Goodwin, J.; and Veale, T. 2012. Full-FACE poetry generation. In *Proceedings of the 3rd International Conference on Computational Creativity*, 95–102.

Díaz-Agudo, B.; Gervás, P.; and González-Calero, P. A. 2002. Poetry generation in COLIBRI. In *European Conference on Case-Based Reasoning*, 73–87. Springer.

Gervás, P. 2000. WASP: Evaluation of different strategies for the automatic generation of Spanish verse. In *Proceedings of the AISB-00 symposium on creative & cultural aspects of AI*, 93–100.

Gervás, P. 2001. An expert system for the composition of formal Spanish poetry. *Knowledge-Based Systems* 14(3):181–188.

Gervás, P. 2002. Exploring quantitative evaluations of the creativity of automatic poets. In *Workshop on Creative Systems, Approaches to Creativity in Artificial Intelligence and Cognitive Science*.

Ghazvininejad, M.; Shi, X.; Choi, Y.; and Knight, K. 2016. Generating topical poetry. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 1183–1191.

He, J.; Zhou, M.; and Jiang, L. 2012. Generating Chinese classical poems with statistical machine translation models. In *Proceedings of the 26th AAAI Conference on Artificial Intelligence*, 1650–1656.

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.

Levy, R. P. 2001. A computational model of poetic creativity with neural network as measure of adaptive fitness. In *Proceedings of the ICCBR-01 Workshop on Creative Systems*.

Liu, C.-W.; Lowe, R.; Serban, I. V.; Noseworthy, M.; Charlin, L.; and Pineau, J. 2016. How NOT to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. *arXiv preprint arXiv:1603.08023*.

Manurung, H. 2004. *An evolutionary algorithm approach to poetry generation*. Ph.D. Dissertation, School of Informatics, University of Edinburgh.

Mihalcea, R., and Tarau, P. 2004. TextRank: Bringing order into texts. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*. ACL.

Mikolov, T.; Chen, K.; Corrado, G.; and Dean, J. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Netzer, Y.; Gabay, D.; Goldberg, Y.; and Elhadad, M. 2009. Gaiku: Generating haiku with word associations norms. In *Proceedings of the Workshop on Computational Approaches to Linguistic Creativity*, 32–39. ACL.

Oliveira, H. G.; Cardoso, F. A.; and Pereira, F. C. 2007. Exploring different strategies for the automatic generation of song lyrics with tra-la-lyrics. In *Proceedings of 13th Portuguese Conference on Artificial Intelligence, EPIA*, 57–68.

Oliveira, H. G. 2012. PoeTryMe: a versatile platform for poetry generation. *Computational Creativity, Concept Invention, and General Intelligence* 1:21.

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. Santiago de Compostela, Spain: ACL.

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.

Speer, R., and Havasi, C. 2012. Representing general relational knowledge in ConceptNet 5. *Proceedings of LREC*.

Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; and Salakhutdinov, R. 2014. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research 15* 1929–1958.

Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; and Wojna, Z. 2016. Rethinking the Inception architecture for computer vision. In *The IEEE Conference on Computer Vision and Pattern Recognition*, 2818–2826.

Toivanen, J.; Toivonen, H.; Valitutti, A.; and Gross, O. 2012. Corpus-based generation of content and form in poetry. In *Proceedings of the 3rd International Conference on Computational Creativity*, 175–179.

Toivonen, H.; Gross, O.; Toivanen, J. M.; and Valitutti, A. 2013. On creative uses of word associations. In *Synergies of Soft Computing and Statistics for Intelligent Data Analysis*. Springer. 17–24.

Wang, Z.; He, W.; Wu, H.; Wu, H.; Li, W.; Wang, H.; and Chen, E. 2016. Chinese poetry generation with planning based neural network. *arXiv preprint arXiv:1610.09889*.

Wang, Q.; Luo, T.; and Wang, D. 2016. Can machine generate traditional Chinese poetry? A Feigenbaum test. In *Advances in Brain Inspired Cognitive Systems: 8th International Conference*, 34–46. Beijing, China: Springer.

Xu, L.; Jiang, L.; Qin, C.; Wang, Z.; and Du, D. 2018. How images inspire poems: Generating classical Chinese poetry from images with memory networks. In *Proceedings of the 32nd AAAI Conference on Artifical Intelligence*, (in press).

Yan, R.; Jiang, H.; Lapata, M.; Lin, S.-D.; Lv, X.; and Li, X. 2013. i, Poet: Automatic Chinese poetry composition through a generative summarization framework under constrained optimization. In *Proc. of the 23rd International Joint Conference on Artificial Intelligence*, 2197–2203.

Zaremba, W.; Sutskever, I.; and Vinyals, O. 2014. Recurrent neural network regularization. *arXiv preprint arXiv:1409.2329*.

Zhang, X., and Lapata, M. 2014. Chinese poetry generation with recurrent neural networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, 670–680.

BISLON: BISociative SLOgaN generation based on stylistic literary devices

Andraž Repar[1,3], **Matej Martinc**[1,2], **Martin Žnidaršič**[2] **Senja Pollak**[2,4]
[1] Jožef Stefan Postgraduate School and [2] Jožef Stefan Institute, Ljubljana, Slovenia
[3] Iolar d.o.o., Ljubljana, Slovenia
[4] Usher Institute, University of Edinburgh, UK
andraz.repar@iolar.com,{matej.martinc,martin.znidarsic,senja.pollak}@ijs.si

Abstract

We describe a novel slogan generator that employs bisociation in combination with the selection of stylistic literary devices. Advertising slogans are a key marketing tool for every company and a memorable slogan provides an advantage on the market. A good slogan is catchy and unique and projects the values of the company. To get an insight in construction of such slogans, we first analyze a large corpus of advertising slogans in terms of alliteration, assonance, consonance and rhyme. Then we develop an approach for constructing slogans that contain these stylistic devices which can help make the slogans easy to remember. At the same time, we use bisociation to imprint a unique message into the slogan by allowing the user to specify the original and bisociated domains from where the generator selects the words. These word sets are first expanded with the help of FastText embeddings and then used to fill in the empty slots in slogan skeletons generated from a database of existing slogans. We use a language model to increase semantical cohesion of generated slogans and a relevance evaluation system to score the slogans by their connectedness to the selected domains. The evaluation of generated slogans for two companies shows that even if slogan generation is a hard problem, we can find some generated slogans that are suitable for the use in production without any modification and a much larger number of slogans that are positively evaluated according to at least one criteria (e.g., humor, catchiness).

Introduction

A slogan is a key marketing asset for any company trying to sell its products and having a good slogan can make an enormous difference on their success. It can drive brand recognition and increase customer loyalty. Slogans are usually produced in brainstorming sessions that involve multiple people. Having a tool that would provide a large set of initial slogan candidates could potentially be of great benefit to marketers and advertisers.

Computational creativity is concerned with machines that exhibit behaviors that might reasonably be deemed creative (Colton and Wiggins, 2012), and our slogan generation system is conceived as a creative system supporting humans in their creative behavior. Closely related research areas include computational humor (Ritchie, 2009; Stock and Strapparava, 2003; Dybala et al., 2010) and poetry generation (see survey of Oliveira (2017)), with the most related to our approach being the lyrics generation system of Bay, Bodily, and Ventura (2017), which transforms an existing text based on certain parameters, including literary devices.

Several approaches to slogan generation have been developed in recent years. BRAINSUP (Özbal, Pighin, and Strapparava, 2013) is an extensible framework for generation of creative sentences. Users can select *target words* that have to appear in the final generated sentence and control the generation process across several dimensions, namely emotions, colors, domain relatedness and phonetic properties (rhymes, alliterations and plosives). The sentence generation process is based on sequences of morpho-syntactic patterns (skeletons) extracted from a corpus of existing marketing slogans. Tomašič, Papa, and Žnidaršič (2015) introduce genetic algorithms to mimic the brainstorming process, while in Žnidaršič, Tomašič, and Papa (2015) they propose a case-based reasoning approach.

We propose a new slogan generation system using literary stylistic devices—that we also analyze on a corpus of existing slogans—in a novel bisociative setting. Koestler (1964) argues that the essence of creativity lies in perceiving of an idea in two self-consistent but habitually incompatible contextual frames, and we use this cross-context approach as a principle in the design of our system. Compared to other slogan generation systems, the input to our system is neither individual words (as in Özbal, Pighin, and Strapparava (2013) or various online slogan generator systems[1]) nor a single document, as in Tomašič, Papa, and Žnidaršič (2015), but the documents from two distinct domains, resulting in bisociative slogans, bearing the marks of both domains.

Outline of the BISLON approach

The main aim of BISLON is to produce innovative slogan candidates of good quality, similar to the ones produced by marketing professionals. Our approach to slogan generation has two principal characteristics. First, it is based on literary stylistic devices (rhyming, alliteration, consonance and assonance) and second, it is related to the concept of bisociation (Koestler, 1964), which has not yet been explored in

[1] E.g., https://www.shopify.com/tools/slogan-maker

the context of slogan generation.

As a resource of slogan skeletons, we use a database of 5,287 English slogans.[2] Our slogan generation mechanism uses a large number of natural language processing (NLP) techniques, and was designed in order to correspond to the following properties of a good slogan:

- For a slogan to be *catchy* and *memorable*, the system uses literary stylistic devices (e.g., rhyme, alliteration).
- For a slogan to be *unique, interesting* and *surprising*, we propose a bisociative slogan generation mechanism. In a very simplified way, we can understand bisociations as cross-context associations and in our system we blend two matrices of thought (two input domains) into a new combined matrix. Word embeddings and Metaphor Magnet metaphores (Veale and Li, 2012) in candidate word generation process also contribute to surprising outputs.
- To address *relatedness* to the domain of interest, the system has a scoring function for weighting domain words.
- For *semantic* and *syntactic* cohesion, we use syntactic skeletons from existing slogans, perplexity computed in relation to the language model and a spell checker.

System Input

The system allows three types of inputs (the last two are optional but advised, in order to increase the variety and the relevance of generated slogans):

- A set of *original* and *bisociated text documents*: To support bisociation, the user is asked to input the documents from two domains. E.g., the user can select one domain as the domain describing the company for which the slogans are generated (*original* domain) and the *bisociated domain* can be selected based on some distant association.
- *Metaphor Magnet terms*: Users can define target and source concepts (corresponding to the original and bisociated input domains).
- *Domain specific terms*: The terms can be either manually defined as keywords of interest or extracted automatically from uploaded documents. We opted for automated term extraction on original documents, using the system from Pollak et al. (2012).

Literary stylistic devices

To enhance memorability and catchiness, slogans may contain various stylistic literary devices, such as rhyme, consonance, assonance and alliteration, which have roots in poetry. According to Baidick (2008):

- **Alliteration** is the repetition of the same sounds—usually initial consonants of words or of stressed syllables—in any sequence of neighboring words. For example, the initial sound L in: ***L**andscape-**l**over, **l**ord of **l**anguage*.[3]
- **Consonance** is the repetition of an identical or similar consonant in neighboring words whose vowel sounds are different, like the consonant K in this sentence: ***D**i**ck** li**k**es his new bi**k**e*.
- **Assonance** is the repetition of identical or similar vowels in the stressed syllables (and sometimes in the following unstressed syllables) of neighboring words: *The engin**ee**r held the st**ee**ring wh**ee**l to st**ee**r the v**e**hicle*.
- **Rhyme** is the identity of sound between syllables or paired groups of syllables, usually at the end of verse lines. For example: *A **t**aste too good to **w**aste*.

For phonetic analysis of the words, we used the NLTK implementation of Carnegie Mellon Pronouncing Dictionary (CMPD) (Lenzo, 2007) which returns a list of phonemes for each word. Take as an example the word "house", for which the following output is obtained: [[HH, AW1, S]].

Analysis of literary devices in existing slogans

In this section we present the analysis of the usage of literary devices in a collection of real slogans. We focus only on nouns, verbs, adjectives and adverbs, since these are the parts-of-speech (POS) replaced in the generation step.

Rhyming

We calculated the rhyming level of two words, i.e. the number of ending phonemes that match, on the basis of the CMPD dictionary. According to Baidick (2008), a phoneme is a minimal unit of potentially meaningful sound within a given language's system of recognized sound distinctions. In general, two words rhyme if they have the same final stressed vowel and all the sounds following it to the end of the word. Example of a rhyme is: *Think about your **car** when you go to the **bar***.

No. of slogans	Precision
123	0.93

Table 1: Analysis of slogans containing rhymes.

As we can observe from Table 1, our system can recognize rhyming with a high level of precision. Our algorithm detects 123 potential rhyming slogans with more than 90% of them being considered true rhymes by a human evaluator. As explained above, we consider only rhymes between nouns, adjectives, verbs and adverbs.

Alliteration, consonance, assonance

For simplified alliteration analysis, we decided to focus only on initial consonants of words. In terms of neighboring words, we included two parameters:

- Strength S denotes the number of words included in the alliteration. If S is set to 3, we get only slogans with more than 3 words in the alliteration sequence.
- D denotes distance between words in the alliteration sequence. If D is set to 1, the distance between the words in the alliteration sequence cannot be greater than 1 (i.e. only one other word can be in between).

[2]We thank Polona Tomašič for her collection of slogans from Internet. Since we do not know how the copyright laws apply to slogans, they are not made publicly available.

[3]The examples in this section are made up due to potential copyright issues.

Consider the sentence: *It **takes** a **true** man to make a **tasty** pie.* If S is set to 3, then this slogan will not be returned by the algorithm. If D is set to 1, then only the first two words (takes, true) will be considered.

In our consonance algorithm, we again made several simplifications: we only focus on identical phonemes relying on the CMPD dictionary and we disregard the second part of the definition about different vowels. Our algorithm essentially detects, within certain parameters, whether words contain the same consonants in non-initial positions (while initial positions are covered by alliteration). Just as with alliteration, the consonance algorithm has the parameters D and S. But unlike alliteration, where good results could be obtained even with a low S, consonance is a subtler device—with the same D, higher values of S are usually needed to produce a pronounced consonance effect.

E.g., consider the two sentences below, where consonance is relatively weak in the first example and very noticeable in the second one.

- A s**l**y and dead**l**y man.
- There is no righ**t** momen**t** to imi**t**a**t**e the beas**t**.

To analyze assonance, we took advantage of the stress annotation offered by the CMPD dictionary to detect only those vowels that have primary or secondary stress. Again, the same two parameters D and S are used. Just like consonance, assonance is also a subtler device than alliteration—for the maximum effect the vowel in question has to be present in several words closely together:

- The m**o**st important man has sp**o**ken.
- Hear the m**e**llow w**e**dding b**e**lls.

We can observe the relatively weak assonance effect in the first example and the comparatively stronger assonance in the second example.

We tested three different configurations of parameters D and S for alliteration, consonance and assonance. In terms of precision, all the results obtained are true examples of the respective literary devices. The only exception are rare cases, where the CMPD returns incorrect pronounciations. For results with different parameters settings, see Table 2.

Configuration	Conf1	Conf2	Conf3
Alliteration	339	11	172
Consonance	106	71	567
Assonance	33	20	222

Table 2: Number of slogans containing alliteration, consonance and assonance discovered with computational means (the actual number of the slogans could be higher). The following parameter configurations were used: D=0,S=1; D=1,S=3; D=10,S=2 for alliteration, D=0,S=2; D=1,S=3; D=2,S=2 for consonance and D=0,S=2; D=1,S=3; D=2,S=2 for assonance.

Compared to the total number of slogans in our database (5,247) the number of slogans found during the analysis is quite low. However, using less limiting settings of parameters would return a higher number of slogans.

Slogan skeleton generation

For slogan generation, the existing slogans in our database are used as the starting point. We converted the slogans to lowercase, tokenized and POS tagged them with the NLTK library (Bird, Klein, and Loper, 2009). We use the coarse-grained universal tagset, as we suppose that grammatical issues would be fixed by the language model. Every noun, adjective, verb and adverb is removed from the slogans, although we keep the information on their POS tags. Every other word type (prepositions, conjunctions etc.) is carried over to the new slogan. This leaves us with a slogan skeleton with empty slots ready to be filled in with appropriate word candidates.

For the replacement, we introduce a bisociation parameter B, which controls the percentage of original and bisociated domain replacement words. If B is 0.5, half of the words come from the original and the other half from the bisociated domain. If B is 0, then all of the words are from the original domain and if B is 1, then all are from the bisociated domain. According to B, the appropriate number of empty slots in the skeleton are randomly chosen and marked as original or bisociated positions.

The next step in the skeleton creation varies according to the chosen literary device described above. For alliteration, consonance and assonance, the user can control the final shape of the literary device with two parameters:

- Distance D controls the distance between words in the literary device sequence.
- Strength S controls the number of words in the literary device sequence.[4]

Let's consider the sentence *Any man looks extreme with XXX shaving cream*, select alliteration as the literary device, the following parameter configuration: B = 0.5, D = 2 and S = 0.5 and the following skeleton:

- Any NOUN VERB ADJ with NOUN VERB NOUN .

Based on D, S and B, the positions to be filled in with the literary device and original or bisociated replacement words are randomly set (replacement positions are numbered from left to right, starting with 1):
literary device positions [3, 5, 6]
original positions [1, 3, 5]
bisociated positions [2, 4, 6]
For rhyme, there are no D and S parameters. Instead, we use the results of the analysis. From the existing slogans we select the ones that contain rhymes and mark the positions of the rhyming words as literary device positions. If we take the same example as before, which contains a rhyme, the literary device positions would now be: [3, 6].

Candidate word pools

Candidate words generation

After generating slogan skeletons with empty slots marked with POS tags, domain and literary device positions, we

[4]Note that opposed to the analysis phase, this parameter has values between 0 and 1 (0.5 means that half of the words should use the literary device).

need to find appropriate candidate words to fill them in.

From the three types of input (documents, Metaphor Magnet terms and domain terms) we generate eight distinct word pools (noun, verb, adjective and adverb pools for original and bisociated domains). In order to do that, we first tokenize and POS tag the input documents and the list of domain terms. Target and source metaphors from the Metaphor Magnet web service are returned in a form of adjective-noun pairs, which are split and assigned appropriate POS tags. Next, we add the resulting words to their appropriate pools, according to their POS tag and domain.

For the optimal functioning of the slogan generation system, the size of all the word pools needs to be sufficiently large, so the system always has enough good candidates for the empty slots in the slogan skeletons. Large input documents would solve this problem but would also make the system user unfriendly, by requiring a lot of effort from the users to gather this large input corpora. Therefore, we expand our word pools by using FastText embeddings (Bojanowski et al., 2016). We loop through the vocabulary of the input text documents and find 15 most similar words for every word in the vocabulary according to its FastText vector.[5] These additional words are POS tagged[6] and also added to appropriate pools.

Candidate word weighting

Ideally, we want our slogan generator to output slogans as relevant to the specified original and bisociated domains as possible. In order to do that, we assign weights to all the words in the created word pools according to their relevance to either the original or the bisociated domain. The calculation of the word weight for a specific word depends on the source of the word. If a specific word does not appear in the input text documents nor in the list of domain specific terms, the weight is automatically 0, since the word is probably irrelevant to the original and bisociated domains. If the word appears in the input text documents, the relevance weight is calculated according to the following formula:

$$w = \frac{input\_document\_freq}{BNC\_corpus\_freq}$$

input_document_freq is the relative frequency of the word in the concatenation of either all the original or all the bisociated documents and *BNC_corpus_freq* is the relative frequency of the word in the BNC reference corpus (Leech, Rayson, and others, 2014). The comparison between the frequency of terms in a domain corpus and in the corpus of general language is a relatively standard approach[7] for keyness calculation, since it rewards domain specific words.

[5]The 1 million word vector model trained on Wikipedia 2017, UMBC webbase corpus and statmt.org news dataset (16B tokens) https://fasttext.cc/docs/en/english-vectors.html is used for the word similarity calculation. The number of most similar words was determined experimentally and assures good functioning of the system even if the input texts are short.

[6]These words do not have any surrounding context, therefore POS tagging here is less reliable.

[7]It is also the underlying principle in the term extraction tool used in the system input phase.

input_document_freq is not available for words from the list of domain specific terms. Therefore it is replaced by a terminology strength parameter *n* (with the default value of 10) in the relevance weight calculation. Finally, the normalized score is calculated for each weight:

$$normalized\_w_i = \frac{w_i - min(w)}{max(w) - min(w)}$$

We use weight normalization to make the relevance weight more easily interpretable by human readers. The final output of the weighting step are two lists of weighted words, one for the words from the original and the other from the bisociated domain pool.

Slogan generation

In the next step, we loop through a list of slogan skeletons and try to generate a new slogan for every skeleton by filling in the empty slots from left to right (this filling order is necessary for the language-model-based semantic and grammatical checks). For every empty slot, we first generate a list of possible word candidates that fit the following criteria:

- The POS tag and the domain is correct. This is done by using only the candidate words from the appropriate word pool. In order to generate meaningful slogans, a sufficient number of candidates is required for every empty slot. The minimum number of candidates that satisfy the criteria is set as a parameter, with the default value of 30. If there are less candidates than the limit for any empty slot, the slogan with the specific skeleton is not generated.
- If the empty slot is marked as a literary device position, the candidate word needs to have the correct phoneme structure for the production of the literary device. This criterion is not applied to the first literary device position in the slogan. Instead, the system remembers only the phoneme structure of the first chosen replacement word and uses it to select compatible replacement word candidates for the remaining positions. As with the previous criteria, here we also enforce the limit of at least 30 appropriate candidates for every literary device position, otherwise the slogan for the specific skeleton is not generated.
- The candidate is a semantically and grammatically appropriate continuation of the preceding word sequence. For this we use a character-aware deep neural language model (Kim et al., 2016) trained on 200,000 randomly chosen articles from the Wikipedia.[8] For training, the vocabulary size was 50,000 words, only character-level inputs were used and the model was trained for 21 epochs. The semantic and grammatical appropriateness criteria is not enforced, if the empty slot is in the position of the first word in the slogan. Otherwise, the language model takes the part of the already generated slogan left of the empty slot and returns probability for each word candidate, that it fills the next position in the sequence. Only five most probable candidates are chosen for the final list.[9]

[8]90% of the data set was used for training, 10% for validation.

[9]Number five was chosen empirically and represents a balance between a variety and cohesion of the generated slogans.

After the filtering described above, we have a generated list of five candidates for each empty position in the skeleton (except for cases when the empty slot represents the first word, then the number of appropriate candidates is not limited). Out of these candidates we choose the final filler word according to the probabilities calculated from the relevance weights described in section Candidate word weighting. The probabilities for every candidate in the list are calculated by dividing the weight of the candidate with the sum of all the candidate weights in the list. Selection according to the computed selection probabilites was chosen, since it contributes to the variety of generated slogans by allowing that the most probable candidate is not always the one selected. On the other hand, the system keeps the slogans relevant by never selecting the irrelevant words (with 0 weight), if relevant candidates are in the list. If there are no such candidates each candidate has the same probability of being selected.

After the slogan is generated, it is first put through a spell checker[10] which tries to automatically remove possible grammatical mistakes. Finally, the semantic and syntactic cohesion check is performed by calculating the average perplexity of the whole slogan by summing the perplexities of all the words in the slogan and dividing the sum with the number of words. Perplexity is a measure of how well a probability model predicts a sample and represents a standard way of language model evaluation. We set the maximum perplexity score of a slogan to 50, which is slightly more than the perplexity of the language model evaluated on the Wikipedia validation set. If the score is above the selected threshold, we assume that this indicates semantic or syntactic inconsistency and the slogan is discarded.

When the empty slot that needs to be filled in is the first word of a slogan, there is no semantic and grammatical continuation or literary device filtering. This means that the only selection criterion is relevance, which causes that some very relevant candidates are selected very often. To avoid the too frequent repetition, we introduce a maximum repetition parameter with the default value of 10. The slogans with the same first word are grouped and if their count is higher than the repetition parameter, the extra slogans with the lowest perplexity get discarded. In this way, we only keep more "original slogans", which have higher perplexity.

Finally, the remaining generated slogans that passed all the tests described above are sorted by their relevance score, which is calculated as a sum of relevance weights of all the words in the slogan divided by the number of words.

Application

To test our generator in a real setting, we have tried to generate a slogan for two Slovenian companies Iolar and Elea IC. The former is in the translation business and the latter primarily deals with construction. For both firms original and bisociated domain documents were defined and terminology was extracted[11] from the original texts.

In the case of Iolar, we used Wikipedia articles on localization and translation memory and one of Iolar's marketing brochures as original domain texts and articles on eagles, Ireland, flight and aircraft for the bisociated domain, since the name of the company *Iolar* means eagle in Irish Gaelic (also a source of inspiration for the company's existing slogan *Flying over the borders*). For Metaphor Magnet generation, we used the phrase *translation is an +eagle* (the + sign limits the search space to positive connotations).

For Elea, the original domain text consisted of the promotional material describing the company, while for the bisociated domain the concept of Eleatics was selected (specifically, the Wikipedia article on this topic). The concept is related to the name of the company and denotes a pre-Socratic philosophy school. For the Metaphor Magnet, we used the phrase *building is a +philosophy*. Table 3 contains the final vocabulary size for individual domains.

	Iolar		Elea	
	Original	Bisociated	Original	Bisociated
Nouns	9,492	7,046	14,164	1,667
Verbs	2,191	1,409	2,884	423
Adjectives	1,926	1,331	2,796	291
Adverbs	572	518	678	161

Table 3: Number of word candidates (word pool sizes) for original and bisociated domains.

We generated slogans for each of the four literary devices. For alliteration, consonance and assonance, the settings D=2, S=0.8, B=0.3 were used, as we aimed for relatively strong literary device effects. D=2 means that the words considered for the specific effect should be relatively close together, and S=0.8 means that the majority of the words in the slogans will be considered for the effect. To simplify, the literary device effects in the resulting slogans will be very strong. The reason for the relatively low value of B is that we wanted to have the majority of the words coming from the original domain, while a smaller number of the words from the bisociated domain contributes to the variety and a unique character of the slogan.

The system generated altogether 1,400 slogans for Iolar (290 with alliteration, 457 with assonance, 413 with consonance and 240 with rhyme) and 811 slogans for Elea (174 with alliteration, 266 with assonance, 255 with consonance and 116 with rhyme). All the generated slogans, together with the human evaluation scores, are made available here: http://kt.ijs.si/data/cc/slogan_generation.zip

Evaluation

The resulting slogans were evaluated for each company. The evaluation criteria, adapted from Özbal, Pighin, and Strapparava (2013), are the following:

- **Catchiness**: is the slogan attractive, catchy or memorable? [Yes/No]
- **Humor**: is the slogan witty or humorous? [Yes/No];
- **Relatedness**: is the slogan semantically related to the company domain? [Yes/No];
- **Correctness**: is the slogan grammatically correct? [Yes/Minor editing/No];

[10] http://pypi.python.org/pypi/language-check

[11] http://clowdflows.org/workflow/5515/

Set	Catchiness		Humor		Relatedness		Correctness		Usefulness	
	Iolar	Elea	Iolar	Elea	Iolar	Elea	Iolar	Elea	Iolar	Elea
Yes	0.152	0.263	0.085	0.218	0.220	0.282	0.345	0.443	0.035	0.068
No	0.848	0.737	0.915	0.782	0.780	0.718	0.527	0.415	0.880	0.877
Minor editing	-	-	-	-	-	-	0.128	0.142	0.085	0.055

Table 4: Evaluation results for top 400 slogans according to the relevance score.

- **Usefulness**: could the slogan be a good slogan for your company? [Yes/Minor editing/No].

For evaluation in each company, the following slogans were selected, based on different criteria:

- Top 100 generated slogans from each literary device ranked according to the system's relevance score (400 slogans in total).
- An additional 12 slogans from each each literary device for inter-annotator agreement (IAA) calculation (48 in total), taken from top ranked slogans.
- 16 generated slogans from each each literary device with the lowest relevance score (64 slogans in total), aimed at evaluating the accuracy of the relatedness ranking score.
- For Iolar, we add 30 real-life slogans of other translation companies, which were available in our slogan database.

For each company, we split the evaluation dataset into 4 annotation sets, with proportionally and randomly selected slogans based on the above selection criteria. For example, each annotator received 124 top scored slogans (25 slogans from each literary device and 24 selected for IAA experiment) and 16 lowest ranked slogans, leading to 140 slogans in each evaluation set. For IAA, two sets of 24 slogans are annotated by a pair of annotators. Finally, three[12] Iolar evaluation sets also contained 10 real-life slogans used by translation companies today.

The sets were prepared for four employees from each company. While for Iolar four employees agreed to perform the evaluation, in Elea, due to time constraints, only two of our contacts were able to perform the task. The other two Elea evaluation datasets were annotated by five persons not employed in the company, but familiarized with the company's professional activities. This meant that we were unable to calculate IAA for one pair of the Elea datasets.

Results

The evaluation results (see Table 4) indicate that slogan generation remains a very difficult task. Nonetheless, more than 10% of the generated slogans can be considered at least partly useful, a similar number can be considered catchy or funny, while around 50% of slogans were evaluated as correct (categories Yes and Minor editing combined). In general, the results for Elea are a bit higher than for Iolar but the large majority of generated slogans are not considered good enough for actual use. However, our goal—to show that at least some actually useful slogans can be automatically generated—was achieved. In Figure 1 we provide a selection of the generated slogans evaluated as useful by the evaluators (for full list visit the url provided above).

[12]One set was evaluated by one of the paper's authors, who was aware of the difference between the generated and real-life slogans.

Iolar

Texts, translations, techniques.
Localization in central Europe.
Multilingual model of meaning.
Translating various cultures.
Localization of life and language.

Elea

Improve your move.
Highway. Holiday. Railway.
Your power is your tower.
Overpasses of creation and information.
Modernized world standards.

Figure 1: A selection of the best BISLON slogans. A total of 69 (Elea) and 76 (Iolar) slogans were deemed useful (Yes and Minor editing) by the annotators.

Comparing the results of the top 64 slogans against the bottom 64 slogans (Figure 2) confirms that the relevance score works as expected (the relatedness is higher among the top 64 generated slogans). However, there is no obvious qualitative difference in other criteria (in terms of catchiness and humor, the bottom 64 seem to be even better).

	Iolar1	Iolar2	Elea1
Catchiness	1	0.792	0.583
Humor	1	0.792	0.583
Relatedness	0.958	0.75	0.542
Correctness	0.792	0.667	0.583
Usefulness	0.958	0.667	0.917

Table 5: Observed agreement of three IAA sets.

Next, we observed the scores with regard to different literary devices. As it can be seen from Figures 3 and 4, for Elea the rhymes have the highest scores, while for Iolar, the best performing device is alliteration (in terms of usefulness).

The chasm that still needs to be overcome in slogan generation is obvious from comparison with evaluation of real life slogans in Table 6. Apart from humor, all other criteria exhibit much higher scores with usefulness exceeding 80%. However, one mitigating factor is that these slogans have

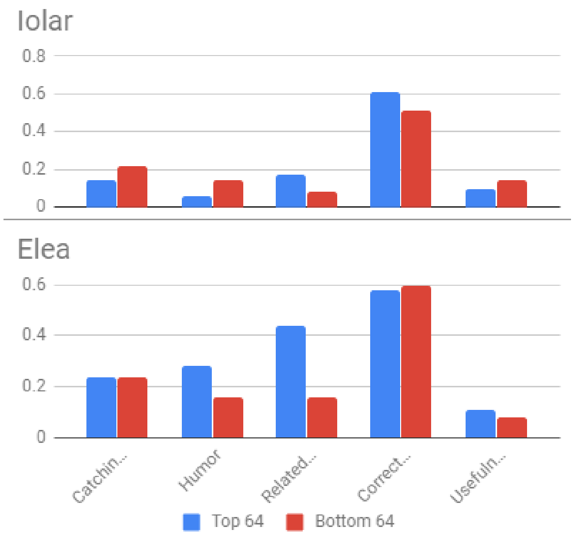


Figure 2: Ratio of positively evaluated slogans among the top and bottom 64 generated slogans (the positive (Yes) and partially positive (Minor editing) scores of the last two categories have been merged.)

most likely been perfected over the course of long brainstorming sessions. It may be more appropriate to compare the BISLON-generated slogans with draft slogans produced during a brainstorming session, to see how many useful slogans are actually generated during such sessions.

Finally, we calculated IAA on three annotation sets - each shared by two annotators. We can see that the overall agreement (OA) (presented in Table 5) is relatively high, with a mean value of 0.942 for the first pair of Iolar annotators, 0.734 for the second pair and 0.642 for the Elea annotators. We also calculated kappa values (Cohen, 1968). Overall, the values are low, but there are differences between the three pairs (average of 0.082 for the first, 0.419 for the second Iolar pair and 0.05 for the Elea pair). Note that the kappa score uses the expected agreement, which is extremely high for the first Iolar and the Elea annotator pairs. As the two Iolar annotators evaluated all the sentences with No for catchiness and humor, the overall agreement is 1, but so is the expected agreement, and consequently the kappa score is very low. According to Landis and Koch (1977), the agreement of the first Iolar annotator pair and of the Elea pair is *low* and the agreement of the second Iolar pair is *moderate*.

	Catch.	Humor	Related.	Correct.	Useful.
Yes	0.767	0.167	0.833	0.900	0.834
No	0.233	0.833	0.167	0.067	0.133
M. edits	-	-	-	0.033	0.033

Table 6: Evaluation results for real-life slogans.

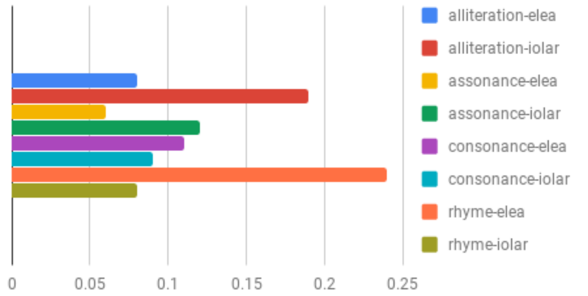


Figure 3: Ratio of useful slogans generated by BISLON (Yes and Minor editing are combined.)

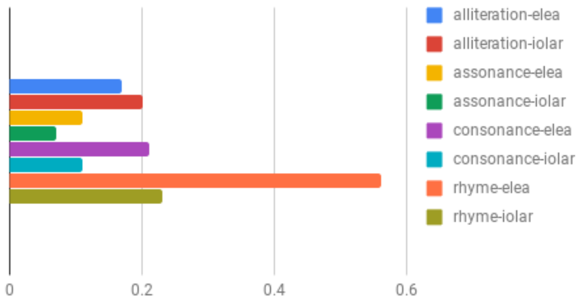


Figure 4: Ratio of catchy slogans generated by BISLON.

Conclusion and future work

This article contains an analysis of marketing slogans in terms of four stylistic literary devices: alliteration, consonance, assonance and rhymes and describes a novel bisociative slogan generator for the four devices. The system has several parameters allowing the user to tweak the strength of the literary device and bisociation and produces a score which ranks the slogans according to their relevance.

The system uses a large number of NLP techniques in order to produce interesting and unexpected slogans. In human evaluation cca. 10% (Yes and Minor editing) of slogans were evaluated as useful, which leaves room for improvement. However, despite the overall low number of positively evaluated generated slogans, the system still produces several good ones, which could actually be used. For example, for Iolar the slogans *Translating various cultures* or *Localization of life and language* are very good candidates. Similar for Elea, which is involved in transport infrastructure construction, a slogan like *Improve your move* could be perfectly applied.

It is hard to judge the actual value of the system. On one hand the results can be compared to other systems for automated slogan generation. The BRAINSUP (Özbal, Pighin, and Strapparava, 2013) framework overall achieves higher results for all the evaluation criteria. We do however believe, that our system could be used more successfully in the early stage of the slogan production process (e.g, in a brainstorming session) since it does not require a very narrowly defined input and produces a large number of very diverse slogans for specified domains, out of which some could be used as out of the box slogans, while others could be used to broaden the space of possible final solutions and discover new meaningful associations. On the other hand, we provided the comparison to human generated slogans. The results for the human generated slogans were significantly higher, which is not surprising, since these slogans are used in production and were most likely already chosen from a list of human generated slogans as best candidates, thoroughly checked and finally approved. Comparing the output of our system to a list of human generated slogan candidates would therefore be a more reasonable comparison and that is something we plan to do in the future, in the context of brainstorming sessions in the advertising industry.

For further work, we plan to perform a detailed analysis of generated slogans, make a systematic evaluation of different parameter settings and, more specifically, analyze the role of bisociation. We also plan to improve several features of the system. First of all, we will analyze the existing slogan database for additional devices used in the slogan production and try to incorporate them into the system. In order to improve syntactic correctness, we will replace the POS tagger using universal tagset by more fine grained tagging or consider incorporating the information from the dependency parser. Semantic cohesion will be improved by training a larger language model and other kinds of semantic features. The system for measuring relevance could be improved by using a more recent corpus to calculate candidate word weight, since some newer words that are not part of the terminology of chosen domains (e.g., download, freeware...) were given very high weights because of their very low frequencies in the BNC. Next, we will allow combinations of different stylistic devices. Finally, a system for sentiment analysis of generated slogans will be implemented. This system will filter out the slogans with negative sentiment, further reducing the number of unuseful slogans. The evaluated slogans will also be considered as a training data for machine learning approaches.

Let's conclude this paper with what was initially a slogan generated for a construction company but is in fact a very wise advice for any situation in life:

The main work also includes the brain.

References

Baidick, C. 2008. Oxford dictionary of literary terms.

Bay, B.; Bodily, P.; and Ventura, D. 2017. Text transformation via constraints and word embedding. In *Proc. of the 8th International Conference on Computational Creativity*, 49–56.

Bird, S.; Klein, E.; and Loper, E. 2009. *Natural language processing with Python: Analyzing text with the natural language toolkit*. O'Reilly Media, Inc.

Bojanowski, P.; Grave, E.; Joulin, A.; and Mikolov, T. 2016. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*.

Cohen, J. 1968. Weighted kappa: Nominal scale agreement provision for scaled disagreement or partial credit. *Psychological bulletin* 70(4):213–220.

Colton, S., and Wiggins, G. 2012. Computational creativity: The final frontier? In *Proc. of the 20th European Conference on Artificial Intelligence*, 21–26.

Dybala, P.; Ptaszynski, M.; Maciejewski, J.; Takahashi, M.; Rzepka, R.; and Araki, K. 2010. Multiagent system for joke generation: Humor and emotions combined in human-agent conversation. *Journal of Ambient Intelligence and Smart Environments* 2:31–48.

Kim, Y.; Jernite, Y.; Sontag, D.; and Rush, A. M. 2016. Character-aware neural language models. In *Proc. of the 30th Conference on Artificial Intelligence*, AAAI'16, 2741–2749. AAAI Press.

Koestler, A. 1964. *The act of creation*. London Hutchinson.

Landis, J. R., and Koch, G. G. 1977. The measurement of observer agreement for categorical data. *Biometrics* 159–174.

Leech, G.; Rayson, P.; et al. 2014. *Word frequencies in written and spoken English: Based on the British National Corpus*. Routledge.

Lenzo, K. 2007. The CMU pronouncing dictionary. http://www.nltk.org/_modules/nltk/corpus/reader/cmudict.html. [Online; accessed 20-April-2018].

Oliveira, H. G. 2017. A survey on intelligent poetry generation: Languages, features, techniques, reutilisation and evaluation. In *Proc. of the 10th International Conference on Natural Language Generation*, 11–20.

Özbal, G.; Pighin, D.; and Strapparava, C. 2013. Brainsup: Brainstorming support for creative sentence generation. In *Proc. of the 51st Annual Meeting of the Association for Computational Linguistics*, volume 1, 1446–1455.

Pollak, S.; Vavpetič, A.; Kranjc, J.; Lavrač, N.; and Vintar, S. 2012. NLP workflow for on-line definition extraction from English and Slovene text corpora. In *Proc. of the 11th Conference on Natural Language Processing KONVENS*, 53–60.

Ritchie, G. 2009. Can computers create humor? *AI Magazine* 30(3):71.

Stock, O., and Strapparava, C. 2003. Getting serious about the development of computational humor. In *Proc. of the 18th International Joint Conference on Artificial Intelligence*, volume 3, 59–64.

Tomašič, P.; Papa, G.; and Žnidaršič, M. 2015. Using a genetic algorithm to produce slogans. *Informatica* 39(2):125–133.

Veale, T., and Li, G. 2012. Specifying viewpoint and information need with affective metaphors: A system demonstration of the metaphor magnet web app/service. In *Proceedings of the ACL 2012 System Demonstrations*, 7–12.

Žnidaršič, M.; Tomašič, P.; and Papa, G. 2015. Case-based slogan production. In *Proc. of the Workshop on Experience and Creativity at the 23rd International Conference on Case-Based Reasoning, ICCBR 2015*. CEUR.

Neighbouring Communities: Interaction, Lessons and Opportunities

Study Paper

Michael Cook[1] and **Simon Colton**[1, 2]
[1]The MetaMakers Institute, Falmouth University, UK
[2]Computational Creativity Group, Goldsmiths, University of London, UK
www.metamakersinstitute.com ccg.doc.gold.ac.uk

Abstract

Building, understanding and sharing software that works in creative spaces is increasingly popular and widespread, with many communities outside of academic research interested in pursuing questions highly relevant to Computational Creativity. We report here on several notable communities in the area: the Procedural Generation Jam, the National Novel Generating Month, the Twitterbot community and the #CreativeAI movement. By studying these communities, we benefit from different perspectives on building creative software, as well as how communities of like-minded people form, grow and sustain themselves. We reflect on these communities as sources of lessons for our field and opportunities for future growth and knowledge exchange, as well as raising awareness of sources of inspiration beyond academia.

Introduction

Groups of developers and artists interested in generative software exist worldwide, and despite a lack of institutional support or funding, many of these groups contain a wealth of ideas and resources, and offer important lessons on building sustainable technological and creative communities. Conferences, journals and seminars are vital ways academics can share ideas and progress, but informal, often online-only, neighbouring communities can be isolated from this process. This makes it hard for their work to reach and influence academic circles, and difficult for us to share knowledge, resources, methodologies and philosophy with them.

We survey here four prominent communities working with AI software for generative purposes. For each, we introduce and explain the origins of the community and how they operate. We discuss commonly-used techniques, highlight prominent examples of work, and comment on the structure and history of the community and how it has developed and grown over time. In many cases, the unique origins and structure of a community is as influential in shaping its work as the technical output or goals. We analyse each community and distil lessons that the Computational Creativity community can learn from to improve its relationship with the general public and other communities, and to improve the research that we do. We discuss how we can spread Computational Creativity research results and methodologies to more people, and in particular what we might need to do to get the communities described here to use our ideas and apply our philosophy. We discuss how we can improve the quality of our community, how we can help each other to do better research and be more welcoming to newcomers. These are all hard things to work towards, and hard for any one academic to do alone, but we believe they can improve the quality of our community, and in doing so, make our field better, help it grow, and produce even better research.

Learning from other communities means accepting that we may need to change, and that the way we do things isn't always right. It also means discovering new opportunities for collaboration, and new people waiting to be exposed to the exciting work that we do. We hope this paper paints an exciting vision of the technical communities which neighbour us, and inspires the Computational Creativity community to try new things, engage with new groups, and continue to evolve and adapt in the future. While journalists, broadcasters and documentary makers have consistently covered Computational Creativity projects/ideas, people across society are increasingly writing about being creative with AI software, and software itself being creative. Moreover, initiatives involving creative software, such as the Dartmouth Turing Tests in Creative Arts (bregman.dartmouth.edu/turingtests), are springing up, often with little or no reference to the results from our field. To stay relevant and grow, we believe it is essential for Computational Creativity researchers to engage with broader communities.

We examine four communities: The Procedural Generation Jam, an annual event whose tagline is 'Make Something That Makes Something'; The #CreativeAI movement, a community of technology enthusiasts and artists who share their experiments, data, code and results with one another; the Twitterbot community, who contribute to an ecosystem of bots on the popular social media site; and NaNoGenMo, an annual event where people write code which generates a 50,000 word novel. The remainder of the paper is organised as follows: first, we step through each community in turn, describing its background, its community, and its technical work. We follow this with a section on lessons we can learn from these communities and what changes we could make to improve the future of the Computational Creativity movement. We follow this by describing various opportunities arising from interacting with the communities around and aligned with Computational Creativity research, and we conclude by reflecting on the future of our field.

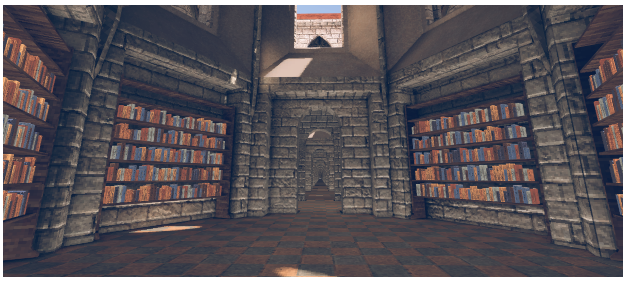


Figure 1: A screenshot from *The Library of Babel*.

PROCJAM

PROCJAM, or the Procedural Generation Jam, is an annual event run around the start of November for nine days. The aim of the event is to 'Make Something That Makes Something', i.e., to make something that is generative. Typically, this comes in the form of software, and because of the event's background, many of the entries are videogames, but PROCJAM benefits from a hugely diverse community that includes artists, crafters, writers, musicians and scientists, as well as game developers. In 2017 PROCJAM had 691 participants, resulting in 174 finished submissions.

PROCJAM's entries are diverse, representing a mix of technical innovation, artistic flourishes, and sharp design. Figure1 shows a screenshot from *The Library of Babel*, an entry to PROCJAM 2015. It recreates the library from Borges' short story of the same name: each book can be opened and read, and contains a randomly generated string of characters, and the library can be explored forever, using a visual trick to generate library rooms as the player moves down seemingly endless corridors. Other entries include *X, a game of Y Z*, which randomly generates chess-like rulesets and lets you play them against an AI, *The Inquisitor*, which simulates a murder and then procedurally arranges evidence and witnesses to let you solve it, and *Dreamer of Electric Sheep*, which uses ConceptNet to create an interactive narrative game where everything is connected by dream logic.

In addition to the event itself, PROCJAM runs several initiatives to build the community and provide resources to people interested in generative techniques. The event has hosted an annual day of talks since 2014, where expert speakers discuss topics related to generativity, including tutorials, surveys and project postmortems. In 2015, PROCJAM began commissioning packs of art designed for manipulation by generative software, and releasing them free under Creative Commons licenses. In 2016, it began publishing an annual zine comprised of community-authored articles about things people had made or discovered in the months between each annual event. In 2017, with funds from its first Kickstarter, PROCJAM was able to pay community members to write tutorials, and awarded a £1,000 support grant to help someone working with generative art.

Techniques

PROCJAM has the shortest official timescale of any of the communities surveyed here, with only nine days to create an entry to the event. PROCJAM does accept late entries at any point, to encourage entrants to take their time, but most entrants stick to this nine-day timeframe which restricts the scope of projects that can be made in that time. This emphasises rapid experimentation with a single technique, rather than the construction of something more complex.

Many entrants use PROCJAM as an opportunity to experience working with generative software for the first time, and will try out some common techniques as part of their entry. Maze generation is a common theme, for example – at least ten entries in 2017 used maze generation, some with interesting twists (such as using computational evolution to evolve harder mazes over time (Ashlock 2010)). Tree and plant generation, often using L-Systems (Lindenmayer 1968), is another popular technique. This can help entrants experiment while getting feedback from an active community, and can inspire new interpretations of well-worn techniques, as people develop them from unique perspectives.

Other entries to PROCJAM leverage more complex emerging technologies, or try out new methods for generating material. For instance, in 2017, there were entries exploring the generation of game rules, which is an active frontier of game AI research (Khalifa and Fayek 2015), and there were also projects using virtual reality. Each year, PROCJAM also sees a number of entries which build on and embellish existing work, such as visualisers for existing generative systems, or extended systems which utilise the output of other generative systems as input for their work. Most jam-style events require entries to be started uniquely for the jam, but PROCJAM encourages existing projects to be extended or reworked, which invites people to perform iterative work as well as breaking ground on new projects.

Community

PROCJAM based its format originally on the popular trend of game jams, but made several modifications to broaden the scope of what could be submitted, and relax the constraints to lower the intensity, e.g., instead of the usual 48 hour timeframe, PROCJAM extended its duration to nine days to allow people to work more slowly, and encouraged late submissions to help people with full-time jobs and children. Inspired by inclusive game jams like Sophie Houlden's Fishing Jam (jam.legendaryfisher.com/), this was well received and broadened participation.

PROCJAM has had widespread impact on both the game development community and the broader generative software community. Nearly 700 people signed up to PROCJAM in 2017, and over 600 entries to PROCJAM have been completed in the four years since it was founded. PROCJAM's site received over 45,000 visitors between February 2017 and February 2018, showing not just the relevance of the event itself, but the contribution the event makes to the community throughout the year, in terms of providing talks, tutorials and resources to people who are eager to learn more about the techniques covered. In addition to this, PROCJAM's video archives on YouTube have over 30,000 views.

The #CreativeAI Movement

'Creative AI' is an overloaded term for several overlapping ideas and communities that exist largely online and keep in touch through social media around the #creativeai hashtag on Twitter (hence our usage of this as a name for the rapid groundswell of international interest). The community is united around finding new ways to use technology creatively, and also democratising the act of building software for creative purposes. Amongst other things, community members discuss: generative AI methods, often with a focus on generative deep learning techniques; exhibitions/concerts/readings/anthologies of material generated by AI systems; where to obtain and how to use AI implementations for creative purposes; and the future of the arts.

Assessing the size of the community is difficult, since there is no single site or collective where the community congregates. There are occasional physical meetings, which can attract up to 100 people, at events such as the London Creative AI Meetup, organised by Luba Elliot (meetup.com/Creative-AI), and members of the Computational Creativity community have been invited to speak at these events. A major aspect of the community involves leveraging new technology for artistic purposes. Various artists emerging from this movement, are beginning to impact the broader art world, e.g., Mario Klingemann (quasimondo.com) has exhibited at the London Photographer's Gallery and the New York Metropolitan Museum of Art.

Through discussions with #CreativeAI members, we have determined that two important pillars of the movement are:

- An emphasis on driving up the quality of the generated outputs to human levels and beyond, with little interest in the idea of the software being co-creative in the process (and often the idea that software could be anything more than a tool is actively disavowed).
- An ultimate aim of mass deployment through commercial level mobile (and other) applications.

The movement may have first coalesced around the popularisation of generative methods that Google's *Inceptionism* (#deepdream) project brought (Mordvintsev, Olah, and Tyka 2015). Moreover, early on in the movement's formation, neural style transfer (Gatys, Ecker, and Bethge 2016) became a popular technique, and the community began exploring ways it could be used to replicate the styles of famous artists, to be transferred onto photographs, drawings, or other works of art. This explorative use of new technology is characteristic of the community, and in the case of style transfer helped popularise the technique. For instance, Alex Champandard's *Deep Forger* (see Fig. 2) provided a public interface to the technology, creating thousands of images and being featured on national news (Champandard 2016).

Another feature of the community is its emphasis on accessible technology – the *Deep Forger* was an impactful project because it enabled people to use neural style transfer without any knowledge of how it worked. Another notable style transfer project which is sometimes referenced under the #CreativeAI banner is the Prisma mobile app (prisma-ai.com). The Prisma app won awards in 2016 on both the iOS app store and the Google Play store, and millions of images have been produced using the app, with the neural style processing being undertaken on servers, rather than on-device. A number of websites for similar kinds of processing are available, such as that at deepart.io.

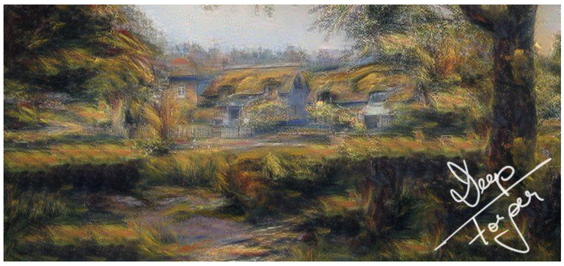


Figure 2: Landscape by Alex Champandard's *Deep Forger*.

Techniques

Much of the most visible work in the #CreativeAI community employs emerging technology whose use and applications may not be fully explored or understood yet. Rather than, for example, pushing the limits of a well-trodden area like evolutionary art, the community is more interested in taking less stable techniques like recurrent neural networks (RNNs) (Sutskever, Martens, and Hinton 2011) and discovering what new domains they can be applied to. This approach can have mixed success, and Twitter is littered with output from RNNs applied to various corpora, with broadly the same outcome as a Markov model. However, the enthusiasm for exploring and experimenting makes this worthwhile, and successes are quickly explored and developed.

#CreativeAI's strongest engagement with the academic community has been through leveraging deep learning techniques, following, and in some cases outpacing, academic communities working on similar topics. A workshop held at the 2017 Neural Information Processing Systems (NIPS) conference, called Machine Learning for Creativity and Design, had a strong contingent of #CreativeAI members, as have previous NIPS workshops on constructive machine learning. Neural networks fit the goals and working style of the #CreativeAI community well – a fast-moving area with a lot of new techniques that are under-explored as the state of the art advances rapidly, providing lots of opportunities to find new applications and uses. The 2017 NIPS workshop included work on photorealistic lip-synch (Kumar et al. 2017), improvisational comedy (Mathewson and Mirowski 2017), story-authoring, anime character generation (Jin et al. 2017), fashion design, fragrance design (Goodwin et al. 2017), and more, showing the breadth of topics tackled by just a small cross-section of the community.

Community

As mentioned above, the #CreativeAI movement is partly defined by how diverse their interests are, which also shows through in the backgrounds of the people in the community. Like many members of the Computational Creativity community, it seems that many #CreativeAI community members combine an interest in technology with an interest in some other creative domain, providing a motivation to find ways to apply new technology, as well as bringing domain-specific knowledge to their work. The community is perhaps

Figure 3: Summary of a game of Botgle, created by the Twitterbot @botglestats and tweeted after the game had ended.

the most industry-leaning of those surveyed here, e.g., the *I'll be Back* series of London meetings brings together generative AI researchers and advertisers. The hashtag #CreativeAI is also linked to job adverts and marketing talks, partly due to it being a general combination of buzzwords, but also because people working in small technology companies make up a large portion of the #CreativeAI community (including the firm Creative.AI, itself a startup company built originally on the same principles as the community).

Twitterbots

Twitterbots are generative programs that automatically post content to Twitter. While the term 'bot' has come to most strongly be associated with malicious intent, the most popular Twitterbots are entertaining or artistic in nature. There is no single organisation or group that creates bots, but one of the larger communities of botmakers go by the label '#botally'. Twitterbots have existed for many years, e.g., *@everyword* by Allison Parrish, one of the most famous bots which tweeted every word in a standard English dictionary, began in 2007, one year after the creation of Twitter itself. Assessing the number of Twitterbots creators is very difficult, but we estimate the size of the community is the largest of the communities described in this paper. For example *Cheap Bots Done Quick*, a website for making bots which we discuss later, has over 7000 registered bots. While users can register multiple bots with the site, this is still a very large number, and represents only a small fraction of the botmaking community.

Twitterbots vary wildly in purpose and behaviour. A popular format for bots is simply to produce a stream of generated tweets at regular intervals, either in perpetuity or until some corpus is exhausted. For instance *@everyword* tweeted each word in its dictionary, in alphabetic (unicode) ordering, once every thirty minutes, until it exhausted its list. Other bots do not have an end point unless APIs change or their creators stop operating them. For example, *@twoheadlines*, by Darius Kazemi, tweets an invented news headline once per hour, using real-world news headlines as source material. In theory, this bot will never stop tweeting.

Twitterbots also exhibit more complex behaviour. For instance, *@botgle* (see figure 3) posts a picture of a Boggle board, a popular word game, every six hours. It accepts entries (in the forms of words players have discovered) for eight minutes, then announces scores at the end. At the end of each month it compiles seasonal statistics based on an aggregate of that month's scores. It has its own dedicated community of players, and a companion bot, *@botglestats*, designed to summarise each game with statistics and notes.

Techniques

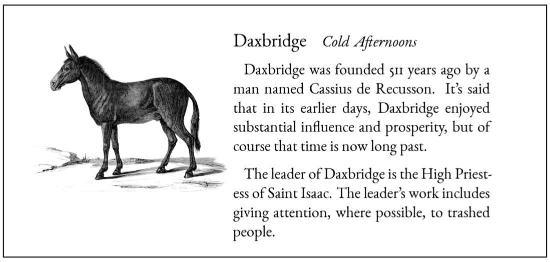
Twitterbot authors use a wide range of techniques for making bots. A popular trend among early botmakers was using Markov models trained on tweets of other users or other corpora. So-called *eBooks* bots became a trend, where people would create companion bots trained on their own tweets. Markov-based approaches work particularly well on Twitter for two main reasons: because the input data is constrained by Twitter's brevity, which makes the resulting Markov model simpler; and because the output is also constrained by the character limit, and Markov models perform better generating short amounts of text, as it hides weaknesses in the model. Lots of small tweets lets the variety of a model show through and reduces the artificiality.

Twitterbots often transcend text, using images or (more rarely) movies as their output. Twitter's multimedia support makes it a platform for the output of bots rather than a medium that they work in, and so it can be more helpful to look at Twitter as a social phenomena for generative software rather than a technical one, which is of great relevance to Computational Creativity, as a field built on its interactions with people (Colton and Wiggins 2012). The most interesting aspect of twitterbots is usually not what they do or how they do it, but how Twitter responds to the bot or how the bot works in the context of Twitter as a social site. For example, *@botgle* (mentioned above) has a huge community of players and bots that work to augment its capability. *@wikisext*, which generates flirty texts mashed up with tutorials from the WikiHow website, replies to users who tweet at it and often gets in long quasi-sexual conversations. Twitterbots are a good example of how the community using the technology shapes how and why things are made.

Community

A major development for the Twitterbot community was the launch of *Cheap Bots, Done Quick* (CBDQ) (cheapbotsdonequick.com), a website for making Twitterbots using the Tracery grammar description language (Compton, Kybartas, and Mateas 2015). CBDQ only requires users to create a Twitter account for their bot and then write a Tracery file describing their bot – no code is written, and no configuration or API access is performed by the user. This makes the creation of a Twitterbot easier, allowing many people to make bots who might not have been able to otherwise.

Although the Twitterbot community is distributed around the world, like many online communities tend to be, it has nevertheless created a culture of sharing and knowledge exchange. Darius Kazemi organised an event called Bot Summit in 2013, 2014 and 2016, in which members of the Twitterbot community gave talks about their approaches, achievements and plans. The 2016 Summit was hosted at the Victoria and Albert Museum in London, and broadcast live online. The community also frequently shares resources


Daxbridge *Cold Afternoons*

Daxbridge was founded 511 years ago by a man named Cassius de Recusson. It's said that in its earlier days, Daxbridge enjoyed substantial influence and prosperity, but of course that time is now long past.

The leader of Daxbridge is the High Priestess of Saint Isaac. The leader's work includes giving attention, where possible, to trashed people.

Figure 4: Emily Short's *Annals Of The Parrigues* (excerpt).

with one another; *Corpora* (github.com/dariusk/corpora), for example, is an online repository of formatted data, such as lists of Greek mythological creatures, occupations, architectural styles and Japanese prefectures. The data is cleanly formatted and consistently organised, despite having over a hundred contributors. There is some overlap of the Computational Creativity and the Twitterbot communities e.g., Michael Cook (first author) and Tony Veale have written and deployed twitterbots, and also have an MIT Press book on Twitterbots forthcoming in 2018.

NaNoGenMo

NaNoGenMo, or the National Novel Generating Month, is an annual event run every November since 2013, with participants aiming to write software which generates a novel of at least 50,000 words. In 2017, the competition had 48 participants. The novel is considered the submission to NaNoGenMo, rather than the software, which is unusual for a software-based jam-style event. The name, format and word limit are all inspired by the National Novel Writing Month, NaNoWriMo, which has run since 1999 and encourages people to write a novel with the same restrictions. NaNoGenMo is a particularly unusual generative challenge because of its scale. A lot of generative projects, including in narrative, focus on smaller target outputs where the creative problems are highly focused. By specifying a minimum size for any entries, NaNoGenMo introduces a new element to the problem, and one of the most fascinating aspects of the event is seeing the unique ways in which entrants tackle this aspect of the task.

A common approach is to subdivide the problem into smaller narratives that can be generated individually and then be sewn into a larger tome. The 2015 entry *The Annals Of The Parrigues* by Emily Short, for instance, uses the structure of a travel guide/travelogue to produce lots of small, self-contained descriptions of places, people, traditions and objects. This use of a pre-existing format that carries with it an expectation of lots of small descriptive text is a highly effective way to break up the 50,000 word task, and the result is of a very high quality. Figure 4 shows an excerpt from the book. Other entries more straightforwardly compose entries from lots of smaller generative works, e.g., the *The Edward Lear Limerick Generator* by Alexander Gladysh is a 54,048 word entry composed of thousands of limericks compiled into a single document.

Another interesting aspect to the event is the emphasis on output, rather than the system. A common question aimed at creative software is how rapidly can it produce things: if we created software that could produce artwork of immense beauty and sophistication, could it churn out such masterpieces on a daily basis? With NaNoGenMo, the emphasis is on a single piece of work. This completely shifts the emphasis within the design of the system. Instead of long-term variety across works, it is more important to have variety within a single work. These small, subtle pressures applied by the challenge result in wildly different approaches to generative systems design.

Techniques

NaNoGenMo community members employ a wide range of techniques based on individual preferences. A partial survey of entries in 2016 found ten languages in use, with Python accounting for 65% of the entries surveyed. Techniques used include Markov models, LSTMs (Hochreiter and Schmidhuber 1997), grammars, cellular automata, and a variety of text analysis techniques (such as word similarity measurement or text summarisation). Famous texts are often used as source material: at least a dozen entries between 2014 and 2016 used Moby Dick in some form, and at least seven used the works of Jane Austen. Often an attempt is made to train systems on text by these authors and then generate new work as a result. In 2017, one project used LSTMs trained on Tolstoy's War And Peace to generate a new novel, and another project attempted LSTM style transfer between War Of The Worlds and Morte D'Arthur.

Some people approach the task of generating a novel with a more artistic interpretation. For instance, the 2017 entry *Pride, Prejudice* by Hugo VK reduced Jane Austen's Pride And Prejudice from 130,000 words to just 51,142 words using a combination of text transformation (contracting phrases like 'will not' to 'won't') and text summary tools to lighten the writing. Another 2017 entry, *The Program Which Generates This Book*, by Martin O'Leary, is a Python program which generates a 58,946 word plain English description of the algorithm which generated the 58,946 word plain English description of itself.

Community

The NaNoGenMo community has considerable overlap with the Twitterbot community (partly because one of its founders, Darius Kazemi, is also a prominent Twitterbot author and community leader). It also has some overlap with PROCJAM, partly through its diverse community, but also because of the chronological overlap – PROCJAM always takes place in November, and encourages submissions of anything generative, resulting in some projects being submitted to both communities. The community features many who are proficient writers and artists in addition to being skilled technically. This leads to a more diverse range of approaches, but also in some cases raises the quality of manually-driven projects by allowing the entrant's own creativity to enhance the work done by the software. Emily Short, an accomplished writer, describes her *Annals Of The Parrigues* as "a story I wrote with the machine", which is evident in the quality of the language and imagery used by the software in creating the finished piece.

Lessons We Can Learn

Having surveyed the communities above, we can draw some general lessons about how they operate, and interpret these in the context of the Computational Creativity, as follows.

Gatekeeping and Accessibility

The communities highlighted above emphasise the openness of their memberships, and a low barrier to entry, which helps people feel more welcome and grows the community faster. By contrast, many academic communities suffer from serious gatekeeping and accessibility issues, some intentional (which we explore in the next section) and some accidental or uncontrollable. For example, there is a general perception that AI research is extremely complex, requiring a lot of education and intelligence to understand. Computational Creativity has its own unique accessibility issues, some stemming from it having been a small, tightly-knit community for a long time. The community has naturally developed its own vocabulary and expectations, which can sometimes drive a wedge between the core community and newcomers.

Unwanted exclusivity can be a difficult subject to accept and address, but having studied these communities, we firmly believe that their low barriers to entry and diverse memberships greatly enhances the work done. Computational Creativity already welcomes a diverse range of academics from a variety of fields and expertises, but we believe that more can be done to open the event up further. This would enhance every aspect of the community, from the kinds of work undertaken, to extending the impact of the work on the wider world.

Increased Sharing of Resources

In the communities described here, there is a big emphasis on sharing resources with one another and creating reusable materials that other people can benefit from. This has many positive aspects: it makes it easier for people to work on the important, novel aspects of their projects instead of focusing on repeating the work already done by others; it also leads to the creation of higher-quality resources over time because multiple people contribute to a single resource. These resources can also be shared beyond our own communities, and end up positively impacting other groups and building bridges between our community and others (we expand on this point in the next section).

One of the great strengths of the Computational Creativity field is the strong and often unique vision many of its practitioners have, and how that manifests in similarly strongly-expressed and unique projects. Uniqueness can have its downsides, however, and this is one reason why it can be hard to break off parts of our work to easily share with others. Previous calls for an emphasis on web services suggest this is an idea that could gain traction (Veale 2013a), and there are examples of useful standalone tools already, but we need to do more to encourage and celebrate this.

One way to achieve this might be to have additional tracks or parts of events like ICCC dedicated to the creation of shared resources, or the pooling of efforts on shared domain problems. Competitions are a good way to achieve the latter – they allow the organisers to set clear parameters for the event, which forces people into similar, if not common, ground. Perhaps hosting competitions similar to a novel generation challenge would encourage people to build new Computational Creativity systems that were all focused on a similar area, which might help produce reusable resources. Competitions for generative systems are not common, but do exist in other areas of AI, for instance the Mario level generation competition (Shaker et al. 2011). Another possibility is that we develop a track for community contributions – useful tools, useful datasets, useful problem benchmarks, open source projects, etc. By highlighting these at our main conference, we not only help promote this type of work, but we also explicitly support and encourage it in future.

Unusual Problem Targets

NaNoGenMo stands out as an event for generative software, in that it produces unusual solutions and has a vibrant community. One possible reason for this is the nature of the event: it takes place over an entire month, and has an extremely specific and difficult goal. While most research work in text generation focuses on shortform writing, NaNoGenMo intentionally sets a much more complex goal. Although these systems may lack the intellect and depth of a system like MEXICA (Pérez y Pérez 2001), conceptually and technologically, the entries are diverse, interesting and thought-provoking. By forcing oneself to aim for something far beyond current capabilities, we reveal new problems, new opportunities, and new ways of thinking about the domains we work in.

While this is not something directly controlled by the community, we would suggest this is something researchers could use to reflect on their own work. For example, all current existing automated game design projects create games which take around 5 minutes to play. What would a system that designed 50-hour games look like? What new challenges would emerge from this new problem setting? What new objectives would it point towards for future work? We are encouraged to think somewhat incrementally as academics, but we can find a lot of rewarding ideas by thinking, at least hypothetically, in terms of larger leaps forward. As suggested above, competitions, or perhaps exhibitions, may be a way to initiate interest around specific new goals.

Opportunities

In this section, we explore some opportunities presented to us by the existence of these communities, to both further the goals of Computational Creativity research, and provide assistance and inspiration to members of these communities.

Expanding the Community

One opportunity presented by these adjacent communities is the possibility to gain new people contributing and attending Computational Creativity events and sharing their knowledge and work. From our conversations with members of these communities, a common perception is that Computational Creativity is hard to break into. This is attributed to many factors, including a concern that newcomers will not know the 'right' papers to cite, and that their work may be judged as being 'merely generative', which is a phrase that has come to strongly divide us from external communities. 'Mere generation' is a particularly unfortunate PR misstep for our community because many of the systems presented at ICCC are, in fact, merely generative. The phrase appears to represent a desire to work for higher goals, rather than a declaration that we are already there, but this is not communicated well to others and often the phrase comes across as dismissive and combative, as pointed out in (Ventura 2016). We believe we need to reassess the role this phrase plays in dictating our relationship with other communities.

Another problem is that many of these communities are not academic in nature, and thus publishing work at a conference is costly and offers little benefit compared to sharing work with an informal online group. This makes it hard to bring people to the conference itself. The Artificial Intelligence in Interactive Digital Entertainment conference (AIIDE) has had success running a Playable Experiences panel in the past (Barot et al. 2015), which invites practical demonstrations of work from outside communities, but this in itself is marred by a lack of travel funds or free time among many of the people in these communities. Ultimately, in order to solve accessibility problems for people outside of academic funding and incentive structures, drastic action would need to be taken that may require funding outside of the reach of a conference like ICCC currently, or an extension of the traditional academic publishing format to incorporate remote attendance or submissions of practical demonstrations as a major part of the conference.

Popularising Computational Creativity Ideas

Many external community member work on projects within the remit of Computational Creativity, or very close to it. Despite this, many people doing this work are unaware of our ideas, or feel unable to apply them. If we can find a way to bridge this gap, we open ourselves up to a potential explosion of innovation and growth for Computational Creativity, a huge wave of potential collaborators, and our ideas finding a strong foothold outside our community. We must accept that for some, building software is enjoyed as a craft exercise, and although we may be eager to share our ideas, it's perfectly understandable that many people will not be seeking them. In particular, while Computational Creativity is often concerned with handing over creative responsibility to software, many people in external communities are interested in producing something beautiful, something personal, or something weird. This doesn't mean their goals are incompatible with ours, but perhaps that we must think about ways our ideas can provide value and interest without forcing people to change their personal goals as developers.

Preparing tutorials or straightforward, practical examples of software which express some of our philosophy may help people grasp our ideas without having to engage with large projects or academic papers, which can be a barrier to entry. For example, when Monte Carlo Tree Search began to become popular in game AI circles (Browne et al. 2012), a website (mcts.ai) was put up to provide understandable working implementations of the technique in common programming languages. Likewise, hundreds of deep learning code repositories are posted on StackExchange yearly usually accompanied by explanatory blog posts. This is much more valuable to an active hobbyist community than links to papers, and we should follow suit with open-source projects demonstrating certain concepts in Computational Creativity such as evaluation or framing.

Another way to popularise our ideas is for Computational Creativity researchers to engage directly with these communities. Such researchers have so far submitted to PROCJAM and NaNoGenMo, made twitterbots and been part of that community, and given talks within the #CreativeAI community. However, it is fair to say that this is not yet mainstream behaviour for Computational Creativity researchers. When we share interests with these communities, we also invite them to learn more about our motivations and where we come from, in much the same way that this paper attempts to illuminate the origins of their communities.

Promoting Our Tools And Resources

Computational Creativity research often grounds itself in the form of bespoke, closed systems, but many web-based tools and other resources have also been developed, such as Metaphor Magnet (Veale 2013b), and FloWr (Charnley, Colton, and Llano 2014). These tools often offer unique functionality or access to unusual datasets, and their web-based nature makes them ideal for use by people who regularly use online corpora like Twitterbot authors. They are excellent ways to promote what we do and who we are, and to positively impact the work done by others.

A productive step here might be to create a community-centric website that lists these tools and resources, with links and explanations of how they work and what they are capable of. The PROSECCO Network website links to a lot of resources like academic papers, including a list of datasets made for the network, but doesn't link to publicly-available web tools or datasets from outside the PROSECCO project. In addition to this, small example projects that use these tools or resources will also help kick-start interest and provide an entry point for less confident people who may still be interested in the possibilities of Computational Creativity.

Conclusions

In the current technological climate, the frontier of artificial intelligence is something academic researchers occupy with corporations, hobbyists and governments. In the last few years, the world has woken up to the idea that software can generate artefacts of real value in truly interesting ways, but people still need some encouragement to explore the potential of software being co-creative or acting as autonomous creative entities. As much as we strive to do research far ahead of the technological curve, we work in an area that is changing rapidly, and changing society with it. It's vital that we look at how other technical communities work so that we can understand how society is making sense of new technology, how we can share our work with the wider world more effectively, and how we can plan for the future of our own community to remain healthy, innovative and exciting.

It is perhaps overly dramatic to suggest that Computational Creativity faces an existential crisis. However, it is worth pointing out a worrying lack of relevance that the field seems to have in other areas of AI research. As a pertinent example, the AAAI paper by Mahadevan (2018) proposed so-called Imagination Machines as a "new overarching challenge for AI", and won a Blue Sky Paper award. Despite covering much-researched topics in Computational Creativity such as metaphor, art generation and ideation, none of the 40 papers it cited were from ICCC conferences, preceding workshops, or current aligned events such as the AISB symposia on Computational Creativity. This is sadly typical of papers covering aspects of creativity coming from mainstream AI and machine learning conferences/journals, which are far more likely to cite work from the #CreativeAI movement than Computational Creativity research (which is normally ignored), e.g., Mahadevan cites work on Creative Adversarial Networks, posted informally on arXiv (Elgammal et. al. 2017). Notwithstanding the very positive reasons given above for engaging with communities of people writing generative software, we should also consider reaching out in order to stay relevant to AI in general.

We have discussed here several major communities working in the space of generative and creative software, describing commonly-used techniques and approaches, the origins and structure of each community, and samples of their work. Some communities are annual events that last for a few days, while others are ongoing groups that are always working and sharing their results with one another, yet they all share a common core of being creative, inventive and interested in using technology for new purposes. We identified lessons that could be learned from these communities, what opportunities they represent for Computational Creativity, and what impact they might have on the future of our field.

We're fortunate to work in a field that is accessible, popular and interesting to the public. It opens up many new opportunities for communication and co-operation that most academic fields can only dream of, but it also means we have to be willing to listen and learn from the wider community, to better understand our neighbours and make the most of our privileged situation. In doing so, we can improve the communities around us, find new places where our work can have impact, improve our own community and stay relevant as Artificial Intelligence ideas and implementations, in particular creative systems, change the world.

References

Ashlock, D. 2010. Automatic generation of game elements via evolution. In *Proc. IEEE Conf. on Comp. Intelligence and Games*.

Barot, C.; Buro, M.; Cook, M.; Eladhari, M. P.; Boyang; Liapis, A.; Johansson, M.; McCoy, J.; Ontañón, S.; Rowe, J. P.; Tomai, E.; Verhagen, H.; and Zook, A. 2015. The AIIDE 2015 workshop program. *AI Magazine* 37(2).

Browne, C.; Powley, E. J.; Whitehouse, D.; Lucas, S. M.; Cowling, P. I.; Rohlfshagen, P.; Tavener, S.; Perez, D.; Samothrakis, S.; and Colton, S. 2012. A Survey of Monte Carlo Tree Search Methods. *IEEE Trans. on Computational Intelligence and AI in Games* 4(1).

Champandard, A. 2016. Semantic style transfer and turning two-bit doodles into fine artworks. *CoRR* abs/1603.01768.

Charnley, J.; Colton, S.; and Llano, T. 2014. The FloWr framework: Automated flowchart construction, optimisation and alteration for creative systems. In *Proceedings of the 5th International Conference on Computational Creativity*.

Colton, S., and Wiggins, G. A. 2012. Computational creativity: The final frontier? In *Proc. of the European Conference on AI*.

Compton, K.; Kybartas, B.; and Mateas, M. 2015. Tracery: An author-focused generative text tool. In *Proceedings of the 8th International Conference on Interactive Digital Storytelling*.

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

Gatys, L.; Ecker, A.; and Bethge, M. 2016. Image style transfer using convolutional neural networks. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*.

Goodwin, R.; Maria, J.; Das, P.; Horesh, R.; Segal, R.; Fu, J.; and Harris, C. 2017. AI for fragrance design. In *Proceedings of the Machine Learning for Creativity and Design Workshop at NIPS*.

Hochreiter, S., and Schmidhuber, J. 1997. Long short-term memory. *Neural Computing* 9(8).

Jin, Y.; Zhang, J.; Li, M.; Tian, Y.; and Zhu, H. 2017. Towards the high-quality anime characters generation with generative adversarial networks. In *Proceedings of the Machine Learning for Creativity and Design Workshop at NIPS*.

Khalifa, A., and Fayek, M. 2015. Automatic puzzle level generation: A general approach using a description language. In *Proc. of the 6th International Conference on Computational Creativity*.

Kumar, R.; Sotelo, J.; Kumar, K.; de Brebisson, A.; and Bengio, Y. 2017. Obamanet: Photo-realistic lip-sync from text. In *Proc. of the Machine Learning for Creativity and Design Workshop at NIPS*.

Lindenmayer, A. 1968. Mathematical models for cellular interactions in development I. filaments with one-sided inputs. *Journal of Theoretical Biology* 18(3).

Mahadevan, S. 2018 Imagination Machines: A New Challenge for Artificial Intelligence. In *Proceedings of AAAI*.

Mathewson, K., and Mirowski, P. 2017. Improvised comedy as a Turing test. In *Proceedings of the Machine Learning for Creativity and Design Workshop at NIPS*.

Mordvintsev, A.; Olah, C.; and Tyka, M. 2015. Inceptionism: Going deeper into neural networks. *research.googleblog.com /2015/06/inceptionism-going-deeper-into-neural.html*.

Pérez y Pérez, R. 2001. MEXICA: A computer model of a cognitive account of creative writing. *Journal of Experimental and Theoretical Artificial Intelligence* 13(2).

Shaker, N.; Togelius, J.; Yannakakis, G.; Weber, B.; Shimizu, T.; Hashiyama, T.; Sorenson, N.; Pasquier, P.; Mawhorter, P.; Takahashi, P.; Smith, G.; and Baumgarten, R. 2011. The 2010 Mario AI championship: Level generation track. *IEEE Transactions on Computational Intelligence and Games* 3(4).

Sutskever, I.; Martens, J.; and Hinton, G. E. 2011. Generating text with recurrent neural networks. In *Proceedings of ICML*.

Veale, T. 2013a. Creativity as a web service: A vision of human and computer creativity in the web era. In *Proc. of the AAAI Spring Symposium on Creativity and (Early) Cognitive Development*.

Veale, T. 2013b. Once more, with feeling! Using creative affective metaphors to express information needs. In *Proceedings of the 4th International Conference on Computational Creativity*.

Ventura, D. 2016. Mere generation: Essential barometer or dated concept? In *Proceedings of the 7th International Conference on Computational Creativity*.

A Parameter-Space Design Methodology for Casual Creators

Simon Colton, Mark J. Nelson, Edward J. Powley, Swen E. Gaudl
Rob Saunders, Blanca Pérez Ferrer, Peter Ivey and Michael Cook
The MetaMakers Institute, Falmouth University, UK

www.metamakersinstitute.com

Abstract

Casual creators are creativity support tools intended to be fun and easy to use for exploratory creation. We have built casual creators called fluidic game designers, which support exploratory game design directly on mobile devices. These work by encoding games in a parameterised design space and enabling player/designers to create new games by varying the parameters in a design interface and seeing changes in their design instantly. A question largely unanswered by existing work is how to choose a suitable parameter space. We describe a methodology for specifying and implementing parameterised design spaces for casual creators, a context that requires balancing a large and expressive space against a manageable and fun user interface. This methodology was derived through investigating and generalising how the parameter spaces for three fluidic games were conceived. It suggests an iterative process whereby parameters are sourced in seven different ways, within a dynamic of incremental expansion and contraction of the parameter spaces.

Introduction

A *casual creator* is design software for creativity support that is oriented towards exploration, by definition enjoyable to use and with a low barrier to entry. Compton and Mateas (2015) describe the act of making with casual creators as an: "intrinsically pleasurable activity, rather than an extrinsically-motivated way to accomplish tasks". Hence casual creators contrast with standard design software such as Adobe's Creative Suite, which typically emphasise large feature sets, often with a complex interface, oriented towards professional productivity. We have been building casual creators that we call *fluidic game* designers, which support exploratory videogame design on mobile devices (smartphones and tablets) in minutes and hours rather than days and weeks, and without a requirement for programming (Nelson et al. 2017a; 2017b; Gaudl et al. 2017).

Our motivation is to bring design of mobile games to the kind of large and diverse audience that characterises mobile-game playing, making casual creators that are more akin to casual games than to professional game-design tools. Fluidic games are designed to reduce the context gap between play and design, by allowing games to be designed on the same devices that they're played on, enabling the user to rapidly alternate between those two modes of game player and game designer, enjoying themselves in both roles.

A casual creator can be seen as an accessible, enjoyable *design space explorer* – a class of design tools that visualise and support navigation of a space of design possibilities supported by the tool, dubbed the *design space* (Woodbury and Burrow 2006a; 2006b). For fluidic games, we use *parametric design* as the technical approach for representing the design possibilities that we support (Woodbury 2010). Parametric design represents design possibilities through an enumerated set of parameters, where each parameter is a possible design choice. Thus the design space is constructed explicitly: N parameters produce an N-dimensional design space, and each design is one point in this space, i.e. one choice of the N parameters. Although computer-supported parametric design is well studied, to our knowledge there is little practical advice on how to construct the actual parameterised design spaces, especially for our purpose of casual creation (versus a context such as engineering optimisation).

Parametric design was pioneered in computer graphics and architecture, where parameters often fall out directly from a choice of "universal" representation, such as a mathematical representation of 3D surfaces. The parameter space then is dictated by this choice of surface representation – for example, choosing a Coons-style (Coons 1967) or NURBS-style (Piegl 1991) surface brings with it a set of parameters. This is also sometimes the case in evolutionary art, where a general representation such as a Compositional Pattern-Producing Network (CPPN) can be used as the basis for design, with parameters falling directly out of the way CPPNs are constructed, as described in (Stanley 2007).

We're sceptical that a useful parameter space for games, especially one for end-user design, can take the form of a similar universal representation. Some parameters will be dictated by underlying technology, such as a physics engine or lighting model, but to cover a meaningful space of casual games requires parameters for characters, player interactions, collisions, dynamics and pacing, visuals, music and audio, scoring, progression and win conditions. Game description languages have identified and formalised many such elements (Schaul 2014; Browne and Maire 2010), but do not in themselves explain how to design a usable parameterised representation. Our main contribution here is to propose a methodology for doing so.

We outline the methods we have used to iteratively build parametric design spaces for fluidic games. The specification of each design space is driven by the twin goals of capturing a meaningful space of games that offers a degree of variety and satisfaction, while providing a simple, enjoyable and comprehensible user interface for design. These two goals are in tension, since expanding creative possibilities requires more parameters and more choices, which produces more complex user interfaces. To address this, our methodology consists of seven different ways to derive parameters within a dynamic of incremental expansion and contraction of parametric spaces in close interplay with the design of user interfaces to navigate them.

To illustrate how and why we arrived at this methodology, we first summarise the development of *Gamika Technologies*, a parameterised game engine. We initially built it by deriving 284 parameters and implementing a user interface called *Cillr* to navigate the parameter space. We describe the parameters and the drawbacks to *Cillr* as an application for developing casual games. Following this, we describe three fluidic games, namely *No Second Chance*, *Wevva* and *Blasta*, which were built by tightly coupling the design of a more focused parameter space with a casual-creator design interface. For each, we describe the derivation of the parameter space and the design of its corresponding casual creator interface, the kinds of games afforded, and some experiments and playtests. This enables us to subsequently present a generalised methodology for parameter-space design of casual creators.

Gamika Technologies

Gamika began as an iOS app for evolutionary art, based on work in (Colton, Cook, and Raad 2011). An initial set of parameters were derived and exposed in a UI to control the art generator. These were then extended to enable the design of *digital fascinators*, i.e., small visual toys designed to hold a player's attention for a few minutes (Pérez Ferrer et al. 2016). We built several prototypes to turn the abstract art pieces into interactive toys, One prototype was a "whack-a-mole" style game, where players tap particles atop the image before they escape. Another was a safecracking game, where the image is split into concentric rings which the player must re-align. A third was a clone of the classic game Breakout, with bricks made from the art.

The fourth prototype, which we called Friends & Foes, emerged as the most promising. This used a feature in the iOS SpriteKit API: converting an image into an object with realistic simulated 2D rigid-body physics based on its contours. In Friends & Foes, the art image is pinned to the centre of the screen, with the player controlling its rotation. Green and red balls (the eponymous "friends" and "foes") spawn at the edges of the screen and are pulled towards the centre. The aim of the game is to have more friends than foes on screen when the timer reaches zero, by using the art image to bat away the foes, but not the friends. Friends & Foes had promise as a game: it requires equal parts tactics and dexterity, and has "hero to zero" moments: one careless move can result in the loss of all the friends accumulated so far. It also depends very strongly on the abstract art image, so multiple levels with varying difficulties were possible. For instance, a rectangular image gives a very different playing experience to a spiral shaped one, which is different again to a smooth circular shape (which makes the game all but unplayable).

Given this promise, in line with producing a casual creator app, we exposed a number of hard-coded values in the software for Friends & Foes as changeable parameters in a user interface called Cillr (see below). In particular, we exposed: the ratio of friends to foes, their spawning rates, speeds and bounciness, and the time limit. These parameters could be tuned differently for friends and foes, so for example the game was made easier by making the foes bouncier (hence easier to bat away) than the friends. To turn this prototype into a game development engine, we continued to identify and expose more parameters, enabling increasingly varied version of Friends & Foes to be made on-device.

One straightforward extension was to allow friends/foes to stick in place after having been in contact with the image for a certain length of time (another parameter). This gives more sense of progress to the game: once a friend or foe is stuck, it cannot be batted away. Interestingly the game with sticking is still recognisably Friends & Foes, but is also recognisably different. This simple change expanded Friends & Foes into a space of two types of games: one with sticking and one without. There are also interesting points at the edges of the parameter space, e.g., if the sticking time is set to zero, balls stick to the image instantly, and the whole strategy of the game changes: foes can no longer be batted away with the image itself, so the player must build a barrier of stuck friends and use that to bat the foes away instead.

To expand the space of games further, we looked at what other aspects of the game could be parameterised, starting with scoring and game ending conditions. In the original game, friends on screen are worth 1 point and foes worth -1, but we extended this to enable scoring when balls stick or are batted away. Also, originally the game ended when the timer ticked down to zero, but we added a parameter to enable the game to end when a certain score was reached, or when a certain number of balls are stuck. We also added conditions for winning or losing based on timing, number of stuck friends/foes and scores. We continued to challenge our assumptions about what aspects of Friends & Foes should be hard-coded and what should be parameterised, e.g., when two balls collide, they were hard-coded to bounce off each other, but we changed that to enable them to stick to each other, and to explode, with a new parameter. Moreover, we enabled collisions to feed into the scoring system, e.g., when a cluster of a certain size forms all balls in the cluster explode, gaining or losing points (reminiscent of the ubiquitous match-3 genre of games). Points could also be awarded or deducted for batting balls off screen, and we added a parameter for the scoring zones.

At this stage, we determined that all of the games in the expanded space were still too recognisable as Friends & Foes, and the casual creator felt like it afforded only versions/levels/variations of this. To greatly increase the range of games, we employed two main tactics. Firstly, we identified some *inspiring examples* as described in (Colton, Pease, and Ritchie 2001), namely some classic casual arcade games that we felt should be in the space of games achievable with Gamika. These included Frogger, Space Invaders and Asteroids. We then determined which parameters would be needed in order for the space of games to include these examples. Secondly, we looked at the underlying physics engine and lighting rendering system available in iOS through the SpriteKit interface, and exposed parameters which were afforded by the methods and fields there.

We also added parameters in ad-hoc ways, including: a systematic search through code to see if any hard-coded values could be extracted; engaging in what-if ideation to imagine different game mechanics; and adding a parameter in order to turn a bug into a feature, or to balance the usage of a different parameter. We describe these sources in general terms when presenting the design methodology below. Note that, in exposing more parameters, we often came up against certain computational limits, so we sometimes restricted the parameter ranges to make it less likely that users could design games with unacceptably low frame rates.

The Set of Parameters in Gamika

After much development, when the parameter-exposing exercise was completed, we had identified and exposed 284 parameters to the user interface for game design. With these parameters, the space of games afforded was sufficiently large to cover radically different games such as four-in-a-row puzzles, bullet hell games and clones of our inspiring examples. Note that one of the computational constraints we imposed was having only two sets of physics objects with a fixed maximum number of each allowed. This did allow us, however, to continue to describe on-screen objects as friends and foes, with the term 'controller' describing the evolutionary art object which the player controls.

The 284 parameters identified for Gamika can be grouped into several categories as follows.

• **Properties** of friends/foes, including their size; shape; colour; sprite image; bounciness, mass and damping.

• **Lighting effects** applied to the background and game objects, controlling: spotlights; ambient light; the calculation of normal maps; and the lit appearance of the friend/foes.

• **Spawning** regimes for the friends/foes. These set: the spawning positions within time-varying ranges; spawn frequencies; total number of each object allowed on-screen; and spatial constraints, such as min/max distances from each other when spawned and spawning on a grid.

• **Movements** of friend/foes, both initially and during a game, via forcefield directions and strengths with parameters for: noise; friction and angular/linear drag; speed limits; whether objects can rotate or not; and how joints such as pins, springs and sliders act on the objects.

• **Collisions** between friends/foes and controller: whether pairs of friend/foe stick, bounce, explode and/or change types on collisions, and timings for these; which screen sides have walls, and how bouncy they and the controller are; and how clusters of friends/foes form and explode.

• **Player interactions** with the controller and friends/foes: tapping, dragging and swiping actions; how the controller is attached by springs, pins and sliders, and can be subject to movement and/or rotation by player touches; tapping to explode, halt, reverse or change the type of the friends/foes; taps on the background spawning more objects.

• **Progress calculations** altering three counters: score, health and lives, via calculations which add up to five measures prescribed by events on friends/foes, which include collisions; explosions; spawning; staying on screen; clusters forming; and objects entering scoring regions.

• **End-game criteria** which dictate how progress calculations and/or game duration terminate the game, which set: what constitutes a win or loss; how the final overall score is calculated; and whether high-scores are recorded.

The Cillr Design App

As mentioned above, *Cillr* was the initial game design interface to Gamika Technologies, an iOS application that enables users to navigate the entire space of games by setting the values for the 284 parameters described above. It was supplemented with additional design screens to (a) save/load games (b) randomly mutate aspects of the game genomes (represented simply as a list of the parameter values) (c) make drawings which get turned into live physics objects, and (d) browse thumbnails of available evolutionary art pieces. Four design screens from Cillr are shown in figure 1, along with some screenshots of example games.

Each parameter in Cillr is assigned a slider in the UI, and these are distributed over screens – which can be scrolled between – with a manageable number on each. In particular, the sliders are grouped into categories with related functionalities to make them more discoverable (the spawning-related sliders are collated, the collision-related sliders likewise, etc.). Even with these groupings, however, we found that having 284 parameters was unwieldy, as even expert users had trouble simply finding the right parameter to change, and it wasn't always clear why a game hadn't changed as expected. We experimented with enabling users to navigate the game space in an evolutionary way, with a screen performing focused random mutations of games. Producing a random game variant, then trying to figure out what it is, can be a fun interaction loop. However, we found that the proportion of playable games produced in this manner was too low to consider it for end-user consumption.

As an initial baseline, *Cillr* is usable, at least by experts. We have used the interface to produce clones of classic games like Frogger, Asteroids and Space Invaders, as well as a variety of novel casual games. Often, we have found that such novel games *emerge* during design, often in response to unforeseen physical interactions of objects. We investigated such emergence with a narrated set of design sessions, as reported in (Colton et al. 2016). Moreover, in a preliminary user test with game-design undergraduate students to test whether *Cillr* could be used as-is, we found them somewhat frustrated by the experience of using it to make games. Interface complexity was one issue, but more importantly, the difficulty of understanding the high-dimensional design space made it hard for these initial testers to grasp what they wanted to do in the app, and how they would begin to do it. Therefore, rather than focusing on improving *Cillr*'s interface, we resigned it to an in-house tool. For public release,

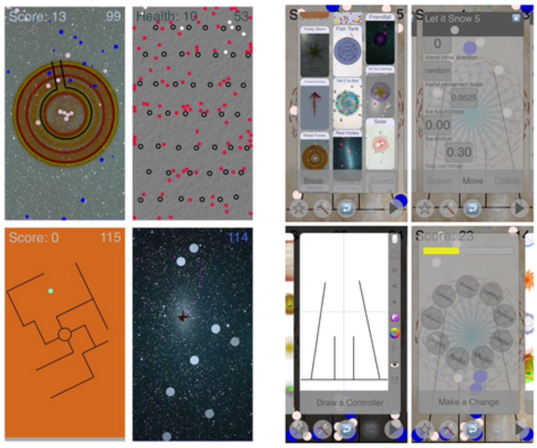


Figure 1: Four Gamika games (left) designed with the *Cillr* app (UI on right). *Cillr* design screens clockwise from top left: List of editable saved games, a screen of movement-related sliders, brainstorming wheel to randomise subsets of parameters, and drawing interface to edit controllers.

we have focused on producing design tools for more cohesive subspaces of parameterised games, as described next.

Fluidic Game Design Apps

With the term *fluidic game*, we mean a casual game for which it is very quick and easy to change any aspect of the design, to improve it, make it easier or more difficult, produce a new level/variation or generally explore possibilities and have fun being creative. As such, an individual fluidic game is itself a design tool, enabling players to search for games, rather than a fixed game in the space. We wanted to blur the line between designing and playing a game, so that it becomes natural for people to change games as they play them, hopefully demystifying and popularising game design, similarly to how level design in games such as Little Big Planet or scene building in Minecraft provide gateways to game design. Hence we give each fluidic game its own user interface, rather than sharing a global one – so that the UI feels like an integral part of the game. In the subsections below, we describe three *fluidic game design apps*, which are collections of parametric fluidic games in a single iOS application, with associated administrative screens, e.g., for collating and sharing games, changing global settings, etc.

In order for it to be as much fun to make fluidic games as to play them, we designed their user interfaces as casual creators, often sacrificing parameters which would increase the space of games to maintain a fun design experience. Despite all being 2D physics-based games, the Gamika space is heterogeneous, with some games puzzle-like, others meditative, and others fast action. Sometimes, changing a parameter will design a slight variant of a game, but sometimes it completely changes the game genre, or could break it. We decided that a fluidic game, by contrast, should encompass a smaller parametric design space that (a) can be navigated more deliberately, with understandable relationships between parameter changes and changes in gameplay behaviour (b) retains emergent properties, so that unexpected

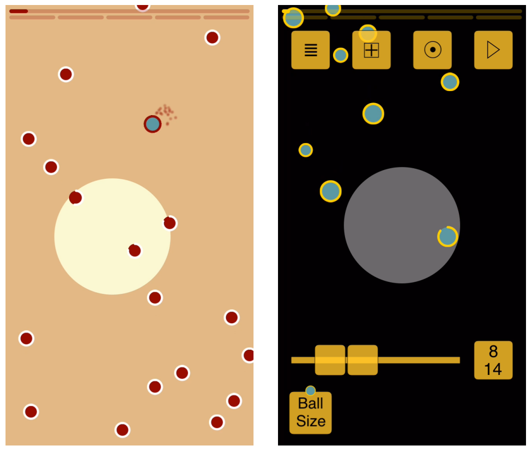


Figure 2: *No Second Chance* gameplay (left) and example design screen (right) for changing the ball sizes.

games can be found, and (c) minimises or eradicates the possibility of producing a broken or slow game.

Having identified a design subspace, it then becomes important to understand the structure of the space well enough to build a casual creator interface that enables designers to control the space's salient features. In the first two fluidic game design apps described below, the parameters shaping the design space are subsets of those of Gamika, previously described in some detail in (Nelson et al. 2017a). In the third app, new parameters have been derived. In each case, the procedure for defining the space of games is based on first cutting down a large set of parameters to a rather small subset, then expanding the set again around the core of a specific type of game. When the expansion happens, this is in close interplay with an interface specific to the genre of game covered by the space. This is followed again by another round of contraction based on user testing with the interface, if this shows that the interface is too complex.

No Second Chance

Using *Cillr*, we designed a game of patience and concentration called Pendulands, where balls (friends/foes) move in a pendulum-type motion and annihilate each other on collision. Players catch balls by hovering under them with a large round target (controller) until they stick. By varying parameters, we discovered that many quite different Pendulands variants could be created. We decided on some fixed elements defining this subspace of Gamika games: the player always controls the target by dragging, and must catch five balls on the target within five minutes. *No Second Chance* is an app built around this space of games. The name comes from a meta-game mechanic: players can share games which are deleted if the receiver doesn't beat the game on the first try (in five minutes). This emphasises the "disposable" nature of games in a generative space, and the challenging nature of figuring out how each one works on first encounter.

The game design screen (see figure 2) is laid out as a hierarchical menu, with submenus allowing visual style and a variety of physics parameters to be changed. As the control and scoring mechanisms are fixed for *No Second Chance* games, new ones are made by varying the nature of movements, collisions and spawning, in addition to visual aesthetics and a soundtrack. Within these constraints, very different types of challenging games can be created. In particular, we found that games requiring gameplay with various levels of *skill* (chasing after balls), *ingenuity* (working out what is going on) and *patience* (waiting for exactly the right moment) were possible. To demonstrate the variety of games that can be produced (and to provide an initial challenge), the app comes with 100 preset games designed using this interface.

To supplement the design interface, we added a 'generate' button, which creates a new game via an evolutionary process. In particular, groups of related parameter settings (e.g., all those for lighting, or movements) from two of the preset games are crossed over and the resulting offspring is mutated. The resulting games are filtered using heuristics to reject clearly bad candidates, and the first four candidates that pass the filter are auto-playtested at super-speed on the device in a split-screen view. As we want games to be playable but not too easy, the app chooses the game that the playtester was able to catch the most balls on, without being able to catch all five. No Second Chance has been tested successfully for game design in an after-school club for 12 year olds, and in rapid game jams (Gaudl et al. 2018).

Wevva

Again using *Cillr*, we made a relatively addictive four-in-a-row game called *Let It Snow*, where snow and rain pour down from the top of the screen (as white and blue balls respectively). When four or more white balls cluster together, they explode and the player gains a point for each, which are then replaced by new ones spawned at the top. Likewise with blue balls, except that the player loses points for them. Players can interact with the game by tapping blue balls to explode them, losing one point in doing so. A grid structure collects the balls into bins, and the best way to play the game involves trapping the blue balls in groups of twos and threes at the bottom, while the whites are exposed above and are continually refreshed through cluster explosions. Occasionally, when all blues are trapped in small clusters, only whites spawn, which looks like snowing (hence the game's name) and is a particularly pleasing moment.

From this single game as a starting point, we first expanded to a larger but still relatively small design space of similar winter-themed puzzle games. These were collected into an app called *Snowfull*, with a fluidic interface allowing designers to change the rain and snow cluster sizes, what happens when the player taps, and the win conditions. A major interface difference from *No Second Chance* was that, given the reduced parameter space, we were able to show a visual overview of all selected parameter values on a single screen, used as (a) an instruction screen for each game, explaining the rules, win conditions, and interaction methods, and (b) the starting point for editing the game.

We conducted a series of 1 to 2 hour rapid game jams using *Snowfull*, with groups from Girlguiding Cornwall (Gaudl et al. 2018). They were able to design games, but gave mixed feedback: despite the casual look-and-feel, the design is oriented towards challenging puzzle-like games,

Figure 3: *Wevva*, showing the overview design screen and a sub-design screen for character movements.

requiring finding a carefully balanced interplay of mechanics that both provide challenge and allow for strategies to overcome those challenges. We found that a large proportion of the participants wanted to make much more casual games, and also felt constrained by wintry theme and the relatively small number of parameters available to change, with game mechanics limited to clustering interaction.

Based on feedback like this, *Snowfull* gradually evolved into a fluidic game called *Wevva*, described in (Powley et al. 2017). While it has many more parameters for game design than *Snowfull*, it is in some ways more causal to design with, and games are certainly more casual to play, although it is still possible to make them challenging. Among other changes, we returned to the idea from *No Second Chance* of the player controlling an in-game object (their avatar of sorts), and also added a number of ways to change sprites, backgrounds and music/audio.

Although we significantly expanded the parameter space, we decided it was important that all selected parameters would be shown in a single-panel visual overview, as portrayed in figure 3 (left). Going row-by-row from top left, the nine grid squares show: character options; what tapping does; physics parameters for character movements; effects of collisions between the avatar and other characters; effects of collisions between non-player characters; music parameters; player avatar look/position, interaction and background; spawning and scoring zones; and win/loss conditions. The constraint that the value of every changeable parameter needed to be represented in this screen was a continual presence in discussions of which parameters to add/remove to expand/contract the space.

Not counting audio design, which is a more advanced topic (as volumes and tempos for five tracks can be set), there are 33 design screens enabling the setting of 47 parameters. We have found that this number seems manageable for most users, and the UI enables rapid progression and a comprehensible, satisfying, design experience. This is achieved through a sensible collection of parameters on screens through which there is a logical progression which rarely gets too far from the home screen. The variety of

Figure 4: *Blasta*, showing the home design screen, a sub-design screen for alien characters, and a game.

games is evidenced by Wevva shipping (on the iOS app store[1]) with 28 games in four different game packs (simple, fast, skilful and tricky). In each of around 15 game jams with Wevva, we have always seen a few genuinely new game mechanics found (and hence novel games). In the most recent tests of Wevva, we found that secondary school children were able to make novel games in around 15 minutes, with zero preparation and no in-game help.

Blasta

A new fluidic game design app called Ludando is currently being developed by the first author as a commercial development for Imaginative AI Ltd. With this app, players will be able to design different types of games including ones in the *Blasta* genre, which covers certain types of shoot-em-ups and driving games. The fluidic games differ to those in Wevva in a number of ways. Firstly, multiple phases (like levels) can be defined for a single genome, so that games can have long gameplay durations, measuring in hours to complete a game. In the design screen, a new phase is constructed by copying the parameters of the previous phase, so it is easy to increase difficulty from level to level. Secondly, the designer can use their own content, namely photographs and audio files in the game. Thirdly, although *Blasta* games differ somewhat by game mechanic, this is not as emphasised as in *Wevva*, and the parameters are instead used to highly fine-tune and personalise the look and feel of standard shoot-em-up games (which have norms and expectations), rather than discovering brand new casual games.

Screenshots of the design UI and an example game are given in figure 4. We see that the home screen breaks down the game design into three main character types (starship, alien1 and alien2) and has design screens for the control mechanisms, the events and the game style. For the two alien types, the sprite and movement patterns for both leaders and wingers can be set, as can parameters for weapons and shields. For the starship, lives, shields and weapons are parameterised, and settings for how the player controls it (by dragging directly, tapping in a desired location or driving it like a car) are also exposed. Style parameters enable the user to set game aspects including backgrounds, terrain, music, animations and the heads-up display.

At the current stage of development, the app has too many parameters, and the interface is useable but somewhat daunting (as per some initial user testing). We are currently reducing the number of parameters, thus limiting the space of games, in order to improve usability of the design interface. *Blasta* was originally intended to cover stealth games, infinite runners and possibly even platform games, in addition to shoot-em-ups and driving games. However, the parameters for these extra genres have been dropped, as the space was too large and the design interface had become unwieldy (much like Gamika). It is likely that some level of homogenisation of aspects like bullets (which can be altered in detail) is still required, and we plan to give designers collections of parameters, e.g., for movement formations, rather than access to the parameters themselves. After the contracting stage, we will undertake substantial play-testing of the app, which will inform more expansion and contraction.

Constructing Parametric Design Spaces

Based on our experiences designing Gamika/Cillr and the three fluidic game design apps above, we have abstracted out a general methodology for building parametric design spaces specifically for use in casual creators. The methodology consists of two parts: (i) sourcing a set of parameters, and (ii) deciding how and when to add or remove parameters, which we suggest can be achieved iteratively through expansion and contraction cycles. Dealing first with the sourcing of parameters, looking systematically at why each parameter in Gamika and the fluidic games was introduced, we have identified the following seven sources:

1. Capture an initial example: Choose one game to implement, and identify the minimum set of parameters needed to represent that game. At the beginning of the Gamika project, we started with a simple game called *Friends & Foes*, which necessitated some obvious initial parameters, e.g., to capture spawning, speeds, etc.

2. Externalise fixed parameters: Systematically investigate values that were hard-coded when implementing the first game – e.g., constants in the source code – and turn appropriate ones into parameters of the design space.

3. Capture an inspiring example: Think of a game that seems possible to express in the current space, and if it isn't, add new parameters to expand the space until either that game, or something like it, is included. For example, in Gamika we expanded the space in an attempt to capture *Frogger*-like games. We didn't end up with precisely *Frogger*, but the exercise helped us identify a number of new parameters that made sense to add to the design space, and we ended up with Frogger-like games in the space.

[1] https://itunes.apple.com/gb/app/wevva/id1322519841

4. Pass through parameters from underlying technology: Study the fields and input parameters to methods available within the APIs available in the programming environment being used, and expose suitable ones as parameters in the design space. For example, Gamika is built on the iOS SpriteKit in which physics objects have a field called *restitution*, which we just passed through directly as *bounciness*. These are exploratory parameters: in contrast to those added to capture a specific example, they're added opportunistically to see what new possibilities they might enable.

5. Balance other parameters: If it becomes clear that having parameterised a game setting X, you really needed to be able to change Y too, then add the extra parameter. For example, in Gamika, we added the ability for objects of the same type to stick to each other, and to explode once a cluster of a certain size was formed. We then realised that we should provide similar functionality for heterogeneous clusters; i.e., clusters of both friends and foes. Identifying balancing parameters can be done systematically post-hoc.

6. Reify emergent features: When experimenting with a parameterised design space, combinations of features often produce novel emergent effects. It may be possible to make an emergent property available in a game, and parameters for turning it on/off and altering it can be added. Doing so makes them easier to use, as the user is now aware that such effects are possible and can control them directly.

7. Split parameters: An existing parameter may be employed for more than one aspect of the game design due to bundling things together in the code, and it may be possible to split out parameters for each aspect. For example, an initial implementation may have a single lighting parameters which controls the overall light level. Later, this could be split into multiple parameters, such as controlling diffuse and specular lighting separately.

The fourth source of parameters above is the one closest to much of the work on parametric design in graphics and architecture, but in our experience, it accounts for only a minority of the parameters required to specify a casual game.

Given these various ways to identify parameters, the other important feature of our proposed design methodology for parameterised design spaces is when and how to introduce or remove the parameters. In retrospect, we can characterise the development of fluidic games (as a form of casual creator) as involving a series of parameter expansions and contractions, carried out in the following four stages:

• Stage 1: unconstrained expansion. First add a large number of parameters, sourced via all seven methods above, to map out as general a space as is feasible. In our case, this is described above in how we arrived at the parameter space for Gamika. This produces an initial parameterised design space containing a large number of highly varied games, but one that is likely to be too large and heterogenous to be a good basis for a casual creator.

• Stage 2: radically cut down the parameter space to just enough to encapsulate a single example game. Choosing one promising game within a large game space enables the building of a bare-bones casual creator UI for designing variants of that game. For *No Second Chance* and *Wevva*, we cut down the parameters in Gamika to those required just for the initial games, respectively *Pendulands* and *Let it Snow*.

• Stage 3: UI-constrained expansion. Initial playtests will likely find that users feel constrained by the small design space and will want to modify more game aspects than those available in the bare-bones interface. At this point, parameters can be re-added to meet user requests, but with a strong constraint from the UI: every new parameter must at the same time fit cleanly into the UI, keeping the resulting interface fun to use, as per the notion of a casual creator. In several cases, we chose very specific UI constraints to further structure this process. For instance, in *Wevva*, the 3x3 grid structure for the design interface, and the requirement that all parameter values be readable on the top-level screen, strongly guided how we added new parameters.

• Stage 4: consolidation and polishing. Even though Stage 3 only adds parameters under a strong constraint of fitting them into the UI, it is possible that the resulting UI may still become over-complicated. It is also likely that the incremental addition process may have resulted in parts of the UI being somewhat inconsisetnt. At this point, relatively modest streamlining can be done by consolidating or linking similar parameters, rethinking arrangements, etc., in preparation for a final version of the casual creator.

Conclusions and Future Work

We have proposed a methodology for specifying parameterised design spaces for casual creators, derived from our experiences in developing three casual creator apps for broadly accessible game design. Our method includes seven sources of the parameters themselves, and a four-stage process of expansion and contraction of the parameter space – initially as a relatively unconstrained design of parameters in the abstract, and later in tight interplay with the design of a casual-creator user interface. We were surprised to find that, although there is a large literature on parameterised design, there is little written about how to specify the parameter spaces themselves, which is obviously key to the process. Therefore, we believe a methodology such as the one here is a useful contribution and may be of benefit to casual creator designers. In future work, we would like to understand how general such a methodology is when applied to other types of casual creators, and to extend the methodology with more automated methods of parameterisation.

This methodology currently focuses on discrete design decisions, such as spawn rate or collision response. In some cases, we might additionally want high-level design parameters that impact many lower-level design decisions. Such parameters are common in generative machine-learning systems. For example, Microsoft's SongSmith musical accompaniment system both data-mines parameters from a corpus and makes a virtue of the hyperparameters required by machine-learning models by exposing them as well (Simon, Morris, and Basu 2008) – for example, Markov-model transition weight was made a parameter and dubbed the "jazz factor". Similar high-level parameters could be used in fluidic games for complex assets such as music, sound effects, and visuals, for which users often want broad stylistic control. We plan to add such high-level parameters to *Blasta*.

An alternative way to support high-level design-space navigation is to allow users to link together parameters with constraints. This is perhaps the primary feature of parametric design in architectural CAD tools (Motta and Zdrahal 1996; Woodbury 2010), which allow users to specify, for example, that a component must be twice as long as tall. The user can then resize the component without having to keep linked quantities in sync. Although designing a casual-creator interface for constraint editing would be challenging, giving users a way to accumulate constraints would allow them to navigate large parameter spaces more efficiently.

As a contribution to Computational Creativity, casual creators add an extra constraint on the design of parameter spaces by requiring that human users of creativity support tools must be able to navigate the parameter space and enjoy doing so. The methodology presented here helps with this tension between supporting breadth of creative expression and designing intuitive, fun interfaces. The main way it does so is by requiring the final parameterised design space to be built up *in tandem with* a casual-creator interface for navigating it. This process, dubbed Stage 3 above, ensures that the parameter space meshes nicely with the UI for navigating it. Theory-based constraints on interface design could also be enforced at this stage, such as limits on the cognitive concurrency of design actions (Kavakli and Gero 2002).

In future work, we plan to look at the Computational Creativity literature to identify further sources for parameter extraction. We note that the first three sources above have some relationship to the descriptive IDEA model (Colton, Charnley, and Pease 2011). The IDEA model describes a computationally creative system able to capture inspiring examples, and then fine-tuning to generalise from them, as the first two stages of developing a creative system; for us here, those are the first two sources of parameters for enabling human creativity in a design space. In addition, the iterative expansion and contraction of a design space bears some resemblance to work on design-space 'sculpting' (Smith and Mateas 2011).

When engineering software for autonomous creativity, many Computational Creativity researchers, including ourselves, have specified a parameterised space of artefacts and enabled software to intelligently search the space. Hence, in addition to usage for implementing creativity support tools, we believe the methodology presented here might have broader usage across Computational Creativity research. We have experimented somewhat, but not yet fully investigated how automated techniques could take advantage of fluidic game spaces to make interesting games. We also plan to explore the possibilities for automatic use of the methodology here, i.e., getting software to create at the meta-level by automatically specifying a parametric design space for creative systems, which we believe would represent a major step forward for Computational Creativity systems.

Acknowledgements

This work is funded by EC FP7 grant 621403 (ERA Chair: Games Research Opportunities). We would like to thank the anonymous reviewers for their helpful comments.

References

Browne, C., and Maire, F. 2010. Evolutionary game design. *IEEE Trans. Comp. Intelligence and AI in Games* 2(1).

Colton, S.; Nelson, M. J.; Saunders, R.; Powley, E. J.; Gaudl, S. E.; and Cook, M. 2016. Towards a computational reading of emergence in experimental game design. In *Proceedings of the 2nd ICCC Computational Creativity and Games Workshop*.

Colton, S.; Charnley, J.; and Pease, A. 2011. Computational creativity theory: The FACE and IDEA descriptive models. In *Proc. of the 2nd International Conference on Computational Creativity*.

Colton, S.; Cook, M.; and Raad, A. 2011. Ludic considerations of tablet-based evo-art. In *Proceedings of the EvoMusArt Workshop*.

Colton, S.; Pease, A.; and Ritchie, G. 2001. The effect of input knowledge on creativity. In *Proceedings of the ICCBR'01 Workshop on Creative Systems*.

Compton, K., and Mateas, M. 2015. Casual creators. In *Proc. of the 6th International Conference on Computational Creativity*.

Coons, S. A. 1967. Surfaces for computer-aided design of space forms. Technical Report TR-41, MIT.

Gaudl, S. E.; Nelson, M. J.; Colton, S.; et al. 2017. Exploring novel game spaces with fluidic games. In *Proceedings of the AISB Symposium on AI and Games*.

Gaudl, S. E.; Nelson, M. J.; Colton, S.; et al. 2018. Rapid game jams with fluidic games: A user study and design methodology. *Entertainment Computing* 27.

Kavakli, M., and Gero, J. S. 2002. The structure of concurrent cognitive actions: A case study on novice and expert designers. *Design Studies* 23(1).

Motta, E., and Zdrahal, Z. 1996. Parametric design problem solving. In *Proc. of the Banff Knowledge Acquisition for Knowledge-Based Systems Workshop*.

Nelson, M. J.; Colton, S.; Powley, E. J.; et al. 2017a. Mixed-initiative approaches to on-device mobile game design. In *Proc. of the CHI 2017 Workshop on Mixed-Initiative Creative Interfaces*.

Nelson, M. J.; Gaudl, S. E.; Colton, S.; et al. 2017b. Fluidic games in cultural contexts. In *Proceedings of the 7th International Conference on Computational Creativity*.

Pérez Ferrer, B.; Colton, S.; Powley, E.; et al. 2016. Gamika: Art based game design. *Art/Games* 1.

Piegl, L. 1991. On NURBS: A survey. *IEEE Computer Graphics and Applications* 11(1).

Powley, E. J.; Nelson, M. J.; Gaudl, S. E.; et al. 2017. Wevva: Democratising game design. In *Proceedings of the Conference on Artificial Intelligence and Interactive Digital Entertainment*.

Schaul, T. 2014. An extensible description language for video games. *IEEE Trans. Comp. Intelligence and AI in Games* 6(4).

Simon, I.; Morris, D.; and Basu, S. 2008. MySong: Automatic accompaniment generation for vocal melodies. In *Proc. CHI*.

Smith, A. M., and Mateas, M. 2011. Answer set programming for procedural content generation: A design space approach. *IEEE Trans. Comp. Intelligence and AI in Games* 3(3).

Stanley, K. O. 2007. Compositional pattern producing networks: A novel abstraction of development. *Genetic Programming and Evolvable Machines* 8(2).

Woodbury, R. F. 2010. *Elements of Parametric Design*. Routledge.

Woodbury, R. F., and Burrow, A. L. 2006a. A typology of design space explorers. *AI EDAM* 20(2).

Woodbury, R. F., and Burrow, A. L. 2006b. Whither design space? *AI EDAM* 20(2).

Issues of Authenticity in Autonomously Creative Systems

Simon Colton[1, 2], **Alison Pease**[2] and **Rob Saunders**[1]
[1]The MetaMakers Institute, Falmouth University, UK
[2]Computational Creativity Group, Goldsmiths, University of London, UK
[3]School of Computing, University of Dundee, UK
www.metamakersinstitute.com ccg.doc.gold.ac.uk

Abstract

As the value of computer generated artefacts increases, we need to question how creative software will fit into human culture, both as creative collaborators and autonomously creative entities. We hypothesise that, in certain contexts where creative software has increased or total autonomy, lack of authenticity will be a major limiting factor for creative systems in the arts and possibly elsewhere. After explaining and motivating this hypothesis, we study notions of authenticity in the human context and use this to raise issues of computational authenticity. We propose a number of ways to address these issues, including influencing public perception of creative software, practical approaches to software recording and using its life experiences, and employing alternative methodologies for building creative systems. We believe that identifying, exploring and addressing issues of authenticity will help to maximise the beneficial contributions that autonomously creative software can bring to society.

Introduction and Motivation

As with many other areas of AI research and practice, there has recently been somewhat of a step change in the quality of artefacts generated through the employment of advanced techniques such as deep learning, and in some cases similar advances in the ease of use in deploying generative systems. As an example, looking at artistic/textural style transfer, the process previously involved writing a bespoke program to apply a certain visual texture to an image, or to generate art in the style of a particular artist or movement. Now style transfer merely involves supplying a style image and target image and waiting while a generative artificial neural network is trained and employed, which captures aspects of both style and target in a new image (Gatys, Ecker, and Bethge 2016), with remarkably good results. While pastiche generation is not normally seen as particularly creative, deep learning researchers are beginning to advance the creative autonomy of their generative systems, for instance through so-called Creative Adversarial Networks (Elgammal et al. 2017), and Computational Creativity researchers are employing such techniques in existing autonomous systems such as The Painting Fool (Colton 2012).

As the ubiquity of creative systems and the quality of their output increases, we have to consider how it will fit into human society. Like any advance in technology or change in societal values, there will be pros and cons to having truly autonomous, creative, software embedded in human culture. We start from the position that the advantages of having such software will far outweigh any difficulties that it would bring. The Computational Creativity community is somewhat split over the question of whether effort should be spent in advancing software to fully autonomous levels of creativity, or whether it would be better to concentrate on building co-creative tools for mixed initiative usage. We focus here on future scenarios where software can exhibit the same level of creative autonomy as people in the arts and sciences. In this context, we question whether opportunities for creative systems (and attendant advantages to human society) will be missed through a lack of authenticity in software, due to fundamental differences between people and machines, including a lack of life experiences to draw upon.

To motivate studying authenticity as an issue, consider the following demonstration, which has been carried out at more than a dozen talks by the first author, known hereafter as the *presenter*. The poem in figure 1 is presented as a well-known piece by female poet Maureen Q. Smith. There is then some joint discussion around what the author may have meant with this short poem about childbirth, for instance: the poem may be about her own experience, as per the 'My boy' sentence; the 'begin again' phrase may have been a reference to a literal beginning (of life), or a re-boot for the family; the 'joy', 'pain' and 'tears' are probably literal; the 'fears' may be about the birth process, but equally about the future of the world the baby is born into.

The presenter then points out that he has made a mistake: in fact, the author was a man, called Maurice Q. Smith. A re-evaluation is then undertaken, with the 'pain' perhaps now being projected, or expressing a painful worry about the

Childbirth

Maureen Q. Smith

The joy, the pain, the begin again. My boy.
Born of me, for me, through my tears, through my fears.

Figure 1: Poem about childbirth used in a demonstration addressing authorial intent and authenticity.

birth process. The presenter then points out that Maurice Q. Smith was actually a convicted child molester when he wrote the poem, and that it was widely regarded as depicting the process of grooming a child. The following re-evaluation highlights the suddenly much darker interpretation of 'Born of me, for me' with aspects such as 'The joy, the pain' becoming disturbing if taken literally, and 'fears' perhaps portraying worries about being caught in a criminal act.

The presenter then says that the poem was actually generated by a computer program using texts about childbirth, and suggest that – with no concerns of an unsettling backstory – another re-evaluation can take place. At this point, the demonstrator attempts to hypothesise what the software *meant* when it wrote about 'joy' and 'pain', but realises there was no further valuable meaning to be gleaned from an understanding of the generative process. Similarly, while the 'tears' are obviously not literal, it's seems impossible to project any further meaning onto the usage of this term. The presenter then invites the audience to question whether the poem has now lost some or all of its meaning, whether – now we can't indulge in projecting thoughts/experiences/intentions onto the author – the value of the poem has decreased or not. There tends to be agreement that people project more value onto the poem when they believed it was written by a person. For full disclosure, the presenter ends by revealing that he wrote the poem for the purposes of the demonstration.

Of course, poems can be – and often are – written, and read from third party perspectives. Well-intentioned ideals such as *Death of the Author* (Barthes 1967) and the *intentional fallacy* (Wimsatt and Beardsley 1954), advocate taking poetry and literature at face value, without inferring authorial intention and actively ignoring any knowledge of the author. It is not clear how easy such ideals are to implement in practice. The childbirth example shows how knowledge of merely the author's name naturally affects the reading of the poem, providing much context. Also, poetry is often enjoyed as a performance art, e.g., in poetry slams, where the personality and intentions of the poet shape the performance, and are explicitly included in commentaries. In a more reasonable appreciation of poetry, where at least some people infer authorial backgrounds and intentions – as highlighted by the childbirth poem – there may be an *uncanny valley* effect (Seyama and Nagayama 2007), e.g., if software autonomously wrote a beautiful ballad about teenage love, audiences would question what the software really knows about this topic, and hence what value the song really has.

We believe that many issues around lack of meaning and authorial intention can be understood via the lens of *authenticity*, or lack thereof, in computational systems. In the next section, we study authenticity from various perspectives in a human context. Following this, we use the study to raise certain issues about software authenticity, and suggest ways to address some of the issues, broadly in three areas: managing public perception of software authenticity; enabling software to use its life experiences in the creative process; and employing alternative methodologies for building creative systems. We conclude by discussing what these issues may mean for the future of Computational Creativity research.

Authenticity in the Human Context

A Million Little Pieces by James Frey (2003) is an autobiography about a struggle with drug addiction and rehabilitation. Published in twenty-nine languages, it has sold over 5 million copies, was on Oprah Winfrey's Book Club selection and number one on the New York Times Best Seller list. In 2006, The Smoking Gun published an article (thesmokinggun.com/documents/celebrity/million-little-lies) claiming that many events in (Frey 2003) had not happened and that Frey had fictionalized his life. The public took this hard: Oprah Winfrey said she felt "duped" and publically rebuked him for "fabricating" and "lying" and the public felt "betrayed" (Wyatt 2006). More than 10 class action lawsuits were filed on the grounds of negligent misrepresentation and consumer fraud, with readers asking for compensation for the time they "wasted" reading a book they thought was non-fiction. Publisher Random House withdrew from a deal with Frey and offered full refunds to readers, and some libraries re-catalogued Frey's book as fiction. Post-2006 editions come with disclaimers by both publisher and author, in which Frey writes:

> "My mistake, and it is one I deeply regret, is writing about the person I created in my mind to help me cope, and not the person who went through the experience. . . . I believe, and I understand others strongly disagree, that memoir allows the writer to work from memory instead of from a strict journalistic or historical standard. It is about impression and feeling, about individual recollection . . . It is a subjective truth, altered by the mind of a recovering drug addict and alcoholic. Ultimately, it's a story . . . that I could not have written without having lived the life I've lived." (Frey 2006, p2)

The debate and strong feelings about *A Million Little Pieces* centre on our modern notion of authenticity. This is an ethical characteristic, an ideal which shapes our worldview – "that one should be true to oneself and lead a life that is expressive of what the person takes herself to be" (Varga 2013, p.5). Authenticity is particularly valued in today's "post-truth" culture where the "fake" can spread widely and impactfully via social media and other channels; where we are urged by the self-help movement to get in touch with our "authentic selves"; and where perception of "brand authenticity" is thought to be the prevailing purchasing criterion of consumer behaviour (Morhart et al. 2015). Varga (2013, p. 5) proposes that we are living in "the age of authenticity", and Wilde (p. 361 of (Lindholm 2013)) writes that:

> " "Know thyself" was written over the portal of the antique world. Over the the portal of the new world "Be thyself" shall be written."

We consider here approaches and responses to authenticity in human creativity, from perspectives of Western philosophy, aesthetics, literature, empirical psychology, consumer behaviour research and cultural tourism. Philosopher of art Denis Dutton (2003) provides the useful distinction between *nominal authenticity* – establishing provenance of an artefact – and *expressive authenticity* – whether an artefact genuinely reflects an author's beliefs and values in a socio-historical context. Here, we focus on the latter.

Acceptable Inauthenticity and Non-authenticity

Socially acceptable levels of inauthenticity in human authors and artists vary, depending on culture, audience, time, and the authors themselves. One of the first English novels, "Robinson Crusoe" was presented as an autobiography of a sailor who was stranded on an uninhabited island in the Caribbean for twenty-eight years. Defoe included the phrase "Written by himself" on the cover page, and wrote – as editor – in the preface to the first edition: "The Editor believes the thing to be a just History of Fact; neither is there any Appearance of Fiction in it." In reality, Defoe was a journalist who never left Europe, although he knew of several real-life survival stories. While Defoe did face some criticism at the time, being called "a false, shuffling, prevaricating rascal" by Joseph Addison (Baker 2009), his pretense did not seem to affect the popularity of the book. This is perhaps because the genre of the novel was newly emerging and rules of convention had yet to be formed. Contemporary critics convey contemporary expectations, such as Nicholson Baker (2009), who calls Crusoe "Defoe's most famous hoax".

Since Defoe, there are many examples of people writing *as though it were written by someone else*, with varying degrees of deception. To our knowledge, there has been no criticism of contemporary author JK Rowling for being inauthentic, although she does present us with several examples of authenticity/inauthenticity in her writing. As an example, her wish to be known by initials was partly so that boys might assume she was male, as she thought they would then be more likely to read her stories (note that JK Rowling has no middle name: the 'K' comes from her grandmother's Christian name). Moreover, after success with the *Harry Potter* books, Rowling adopted the non de plume Robert Galbraith, and further invented an appropriate persona and a fictional biography directly related to Galbraith's story topics, adding depth and credibility. This was presented on "his" early books as fact, e.g., the author's biography on the inner sleeve of (Galbraith 2013) reads:

> "After several years with the Royal Military Police, Robert Galbraith was attached to the Special Investigative Branch . . . He left the military in 2003 and has been working since then in the civilian security industry. The idea for Cormoran Strike grew directly out of his own experiences and those of his military friends who returned to the civilian world. 'Robert Galbraith' is a pseudonym." (Errington 2017).

Both of these aspects seem to be acceptably inauthentic, possibly due to Rowling's popularity, reputation, longevity and backstory, and to the fact that she made no further pretense that these were Robert Galbraith's experiences. This is in contrast to Frey, who was unknown before his book, and who promoted it with public appearances in which he propagated the persona he had created (albeit unconsciously). Of course, writing about wizards and witches might also have laid Rowling open to accusations of inauthenticity, but as she doesn't claim to have experience of them, there is no deception, hence this is better described as "non-authentic" behaviour, i.e., not on the authenticity/inauthenticity scale.

As another example, Mark Haddon, in his book *The Curious Incident of the Dog in the Night-time*, took a more transparent approach than Galbraith. The book was written from the perspective of a 15 year-old mathematician with behavioural difficulties. Haddon himself has none of these characteristics, but he escapes charges of inauthenticity by being completely open about who he is. Thus he also falls into the realm of the "non-authentic". These examples show different levels of fiction, ranging from the realistic fiction of Defoe's, Galbraith's or Haddon's books – stories that *could have* occurred, resembling real life, believable settings and characters, to the fantasy of the Harry Potter series – a form of speculative fiction, which also includes science fiction, superhero fiction, science fantasy, horror, alternate history and supernatural fiction. Such a scale may have implications for authenticity judgements, for instance with greater expectations of authenticity in realistic fiction than in speculative fiction, especially if a fictionalised world differed significantly from standard human experience.

The Problem with Experience, Memory and Self

Whatever the level of fiction, boundaries between fiction and reality are impossible to define. No fiction is entirely fictional, as materials are taken from reality. Conversely, reality is not all that solid, there is no formal, ideal perception and any representation involves a point of view, a perspective, criteria of relevance, an implicit theory of reality, and so on. Even if there were an ideal perception, creative representations are usually based on memory, which – as argued by Frey in his disclaimer above – is malleable and unreliable. Experimental psychologist Elizabeth Loftus and colleagues have shown that memories can not only shift, or be lost via decay and repression, but that new, false memories can be implanted, or "recovered" through therapy. She writes: "Only the flimsiest curtain separates reality from imagination"; "The detail people confabulate and then believe in just astounds me"; "What we access is halfdream, half-construct, entirely unreliable" and describes artefacts based on such memories as: "authentically inauthentic" (Slater 2004, Chap. 8).

Loftus shows us the fragility of the connection between our experience and our awareness of experience. This connects to a model of authenticity from person-centred psychology, which Wood et al. (2008) describe and use to develop an authenticity scale questionnaire, designed to show degrees of three different aspects of authenticity. These concern: (a) primary experience, involving physiological states, emotions and cognition; (b) awareness of experience, and (c) behaviour and emotional expression. The first aspect of the model of authenticity is called *Self-Alienation* and describes the relationship between (a) and (b), concerning mismatches between actual experience and conscious awareness of it. The second aspect concerns the relationship between (b) and (c), called *Authentic Living*, and involves congruence between experience as consciously perceived and expression of emotions. The third aspect, *Accepting External Influence*, involves the influence of social environments on both the first and the second aspects, as accepting influence of other people and conforming to expectations of others. Wood et al. propose evaluating each authenticity aspect through a set of similarly phrased statements, with which participants express agreement or disagreement using a Likert scale, from 1 *does not describe me at all*, to 7 *describes me very well*. *Self-Alienation* is judged via statements like "I feel out of touch with the real me", *Authentic Living* via "I always stand by what I believe in", and *Accepting External Influence* via "Other people influence me greatly".

Varga (2013, page 61) elaborates our sense of self, assumed in the *Self-Alienation* aspect of Wood et al. (2008), describing two competing models. The *Inner Sense Model* assumes that the inner self is something which is stable and given, that can be known and expressed. The *Productionist Model* on the other hand, takes the position that there is no fundamental, unchangeable self comprising 'psychological DNA' to discover, but rather we continually create and re-create a dynamic, fluid self in different contexts. These ideas build on philosophical notions of personal identity, which explore issues such as defining and undefining features, changing and contingent defining features, personal identity over time, how we know who we are, how disunity of consciousness or split personalities affect our notion of synchronic identity and what is important when we think of a self. The question of what and who a self is, is important if by authenticity, we mean being true to ourselves.

The existentialist movement in philosophy has elaborated the notions of self and authenticity, seeing the experience of authentic feeling as validating our existence. Much of this was a response against excessive social influence, mannered falsity, refinement and hypocrisy – the *Accepting External Influence* aspect of the person-centred model of authenticity. Rousseau complained about people striving to impress each other and only being able to experience themselves as reflected in the eyes of others, being proud if admired, self-despising if held in contempt, etc. Nietzsche also reacted against his bourgeois, Christian upbringing, in which "no one dares to appear as he is, but masks himself as a cultivated man, as a scholar, as a poet, as a politician" [see page 383 of (Lindholm 2013)], developing the ideal of expressive authenticity and urging people to be themselves. Heidegger wrote about the inauthentic self, which "lives in subservient and unconscious relation to the anonymous and ubiquitous 'they'" [page 385 of (Lindholm 2013)]. Sartre further developed ideas of inauthenticity and bad faith, where identification with social roles, for instance that of a waiter, kills any possibility of authenticity. He writes: "I had been convinced that we were created for the purpose of laughing at the act we put on for each other" [page 387 of (Lindholm 2013)]. These ideas also have found traction in psychotherapy.

Expressive Authenticity in Political Sub-cultures

Expressive authenticity is an important concept in many domains. For instance, immersive musical sub-cultures such as the punk, goth and hip hop communities sometimes dismiss "part-time wannabes" or "hangers on" as poseurs, people assuming a persona in order to be accepted by or seen as members of a group, but who do not understand the group's values or philosophy. Here, we see a collective "self" in terms of authenticity, with members protecting their identity and valuing their authenticity, sometimes to the extent that people who do not share their collective experience are derided for producing – or even listening to – a particular style of music (Jacobson 2018). Personas of "poseur" musicians are sometimes created or elaborated by others, e.g., record companies and trade magazines promoting music stars. This highlights that a key factor in people's willingness or lack thereof in listening to an "inauthentic voice" is politically motivated. Someone from a privileged background, possibly capitalising on or stereotyping a politically marginalized group, may be far less palatable than the reverse. This accounts for the outrage at the James Frey example and possibly for any distaste or discomfort that neurotypical Mark Haddon wrote about a protagonist with autism.

Conversely to the importance of expressive authenticity given to the musical sub-cultures described above, some communities prefer to present artefacts as self-contained, stand-alone objects, independent from their creators. Presenting them alongside a view of the creator is thought to constrain the way in which they are interpreted and understood, putting artificial limits on the artefact. These ideas are advocated by New Criticism, a movement in literary theory, and Roland Barthes in his essay on "The Death of the Author" (Barthes 1967). Since we cannot understand authorial intent, and if we could, it would only limit our reading of their work, the argument is that literary fields should move closer to scientific or engineering fields in how they view creators. Once creators have brought an artefact or idea into being, their role is complete, and the artefact must stand alone and either work or not, much as we might judge a pudding or a machine (Wimsatt and Beardsley 1954, chap 1).

Brand Authenticity

Morhart et al. (2015) considered brand authenticity in the context of consumer behaviour, drawing on the literature and conducting exploratory interviews to develop four dimensions of perceived brand authenticity (PBA): continuity, credibility, integrity and symbolism. Here, continuity relates to longevity and persistent values; credibility to a brand's willingness and ability to deliver on its promises, which Morhart et al. conceptualise as transparency and honesty; integrity to virtue reflected in the brand's intentions and in the values it communicates; and symbolism to brands that reflect values that consumers consider important and that help to construct their image of themselves. Regarding authenticity, Morhart et al. summarise: "PBA is the extent to which consumers perceive a brand to be faithful toward itself (continuity), true to its consumers (credibility), motivated by caring and responsibility (integrity), and able to support consumers in being true to themselves (symbolism)."

The notion of brand authenticity is connected to cultural tourism, which is also heavily invested in the notion of authenticity. "Staged authenticity" (MacCannell 1973) consists of packaging or performing a cultural event in such a way that it conforms to expectations of an authentic tourist experience. The choice of venue and the surrounding context will all contribute to perceptions of authenticity as a consequence of the experience. Alternatively, cultivating a Benjaminian 'aura' (Benjamin 1968) through 'distance' can enhance perception of authenticity in the arts. The emergence of authenticity as proposed by Cohen (1988) suggests that the inauthentic can become authentic over time, possibly as a consequence of the evolution of old traditions, or the establishment of new traditions. As Glaveanu (2017) points out, traditions are not fixed, but amorphous and shifting in response to cultural change. Finally, as Wang (1999) notes, the use of traditional practices creates a link to the past, relating the creative activity to the self, society and world.

Summary of Discussion

Up to this point, writings on authenticity have solely focused on its meaning in a human context. We have seen that:

- Evaluations of authenticity and its importance change over time, given different actors. Today, we may be more generous or forgiving in our authenticity evaluations of authors and artists who have built up popularity, reputation, longevity and backstory.
- Inauthenticity can take the form of obfuscation of identity (cf JK Rowling introducing ambiguity over her gender).
- Writing as though you were someone else can be acceptable. Acceptability is greatly enhanced if the author is transparent about it, where levels of transparency vary from producing a detailed fictional backstory and that role (possibly considered to be unacceptably inauthentic), to a non de plume and un-detailed persona but no accompanying appearances (possibly considered to be acceptably inauthentic), to complete transparency (non-authentic and acceptable).
- In fiction, it *may* be easier to avoid charges of inauthenticity if writing in "non-authentic domains", such as speculative, rather than realistic fiction (although there are contexts in which this wouldn't be the case).
- The notion of "being true to oneself" is psychologically and philosophy very difficult.
- There is a strong connection between life experience and behaviour or artefacts in expressive authenticity. In person-centred psychology, authenticity is described as the connections between *(a)* experience, *(b)* awareness of experience, and *(c)* behaviour which reflects that awareness, and the degree to which external influence affects these connections.
- The self can be seen as stable and knowable (discovered), or as dynamic and context-dependent (invented).
- In highly politicised domains, such as some sub-cultures of music, expressive authenticity is highly prized. In other domains, such as science, maths, and some literary and artistic approaches, authenticity is seen as less important.
- Perceived brand authenticity can be seen in terms of continuity, credibility, integrity and symbolism.
- Perceived authenticity of cultural events and artefacts can be affected by the way in which they are presented.
- Maintaining distance between artist and audience can build auras which might enhance perception of authenticity.
- The perceived authenticity of an artefact can change over time, as the cultural tradition and socio-historical context in which it exists also changes and evolves.

Dealing with Issues of Authenticity

The study above gives us a firm foundation on which to highlight issues of authenticity in software, and to suggest potential ways for dealing with valid related criticisms concerning autonomous software creativity. Anthropomorphisation has been both beneficial and detrimental to Computational Creativity researchers in presenting their work: on the one hand, it certainly helps to describe creative software in terms of human creative processes and products, but on the other hand, it can give a false impression of humanity when there is none. We have argued in (Pease and Colton 2012) that Turing-style tests can be detrimental to the bigger picture of embedding creative software in society. Moreover, in (Colton et al. 2014), we point out that – rather than levelling the playing field as hoped – such tests can actually serve to emphasise a *humanity gap*, i.e., people like an anonymised artefact because they make a human connection, but this is disappointingly removed on revealing that an artefact was made by computer, leading people to realise the implicit expectation of human creativity in making certain forms of art, such as poetry.

Addressing why such a humanity gap may be disappointing, we hypothesise that it is a perceived lack of authenticity as per some of the contexts given above. This can raise a dilemma amongst people appreciating computer-generated art: they may want to express dislike of a piece because of lack of human authenticity, but that may offend their liberal sensitivities in the context of the *Death of the Author* idealogy described previously. That is, they may feel that they are pre-judging software unfairly in a way similar to racism or sexism, and it may not be clear to them whether this is a sensitive issue when applied to machine creativity. As more people are exposed to more high-quality computer generated artefacts, we believe this issue will become more pressing.

We advocate managing people's expectations of forming human connections when presenting computer generated material for cultural consumption, similar to someone being clear they have purchased an e-book rather than printed book for someone (Colton et al. 2014). That is, by escshewing Turing-style tests, getting software to frame its work, and being clear about the computational origins of generated artefacts, it seems possible to present computational creativity as being non-authentic rather than inauthentic, borrowing the terminology from the previous section. This could be taken further, i.e., by enabling software to **own its non-authenticity**, by it being clear that it doesn't have the relevant life experiences to bestow authenticity onto its process and product. The software could then suggest that audience members read/view/listen to its output as if it were created by a particular type of person, e.g., a teenage boy, or a particular (human) individual, etc.

Owning non-authenticity is a short-term possibility for side-stepping issues of authenticity. Another possibility is to **emphasise the product**, e.g., get the software to work in domains where product is far more important than process, e.g., scientific discovery. Put bluntly, if software invents a new cancer drug, no-one will care that it hasn't lost a relative to the disease. In the arts, abstract art is often perceived more as an invitation for a viewer to self-reflect than to interrogate authorial intention, and here, authenticity may be less important. Moreover, as discussed above, authenticity may be less of an issue if, rather than working on realistic fiction, which may need human authenticity to support it, software instead **produces speculative works**, such as science fiction, which was the approach with The WhatIf Machine project (Llano et al. 2016). We note that even here, there may be contexts where authenticity would be paramount.

Referring to brand authenticity, we note that the authors of systems such as Colton's *The Painting Fool* or Pérez y Pérez's *Mexica* have strived to **build a brand** for their programs, by: naming their system; developing it over a long period, collating and celebrating outputs as in (Perez y Perez 2017), writing a plethora of research papers, substantial public engagement, popular press coverage, etc. The notion of an 'aura' around art works and artists is a well known concept (Benjamin 1968), and it's not impossible to imagine software having such a reputation, which could be used to add authenticity to its practice and products.

Ultimately, a lack of life experience of concepts such as love and scenarios such as childbirth leads people to projections of inauthenticity onto software when they create artefacts addressing such things. Software does, however, have life experiences, but not those that people have. For instance, The Painting Fool has interacted with and painted portraits of around 1,000 people, including a few famous people, in multiple countries. It has made people laugh, caused excitement, disappointment and interest and been written about by scores of journalists. One practical way of addressing issues of inauthenticity, is for the software to record and **use life experiences** of this nature in its creative process. That is, the software could record aspects of its creative process, outputs, public and private engagements, then refer to this data in future projects.

While it may be difficult to convince audiences to see the world from the software's perspective and that such computational life experiences are worth celebrating artistically, such an approach wouldn't suffer from being seen as inauthentic. However, the programmatic origins of the software may throw up two difficulties in the general acceptance of the notion that software has its own authentic life experiences. That is, being an engineered software entity may make it difficult for people to (a) project something akin to creative personhood onto software, and (b) empathise with something which is very different to people. It appears that we need to engineer software which moves away from the humanity of the programmer, while simultaneously moving towards the humanity of its audience members.

The *Lovelace Objection*, as framed by Turing (1950), gets to the heart of the issues surrounding the perception of authenticity, i.e., to an onlooker, the productions of a computational system may appear inauthentic because it is natural to look to the programmer as the source of authentic experiences of the world. Turing's response to the objection is remarkable for the scope of his vision of creating 'child machines' with a small set of core features, such that they can be educated (Turing 1950). Consequently, tackling issues of authenticity in autonomous creative systems may require researchers to rethink their role.

Grounding Computational Creativity

The question of authenticity is tied to the question of whether a creative system that is not *grounded* (Brooks 1991) in its world can produce anything authentic. Situated cognition argues that all knowledge is situated in activity bound to social, cultural and physical contexts, and hence that cognition is inseparable from action (Clancey 1997). Embodied cognition argues that many features of cognition, including high-level representations and reasoning (Lakoff and Johnson 1999), are shaped by aspects of the physical body of the agent (Anderson 2003). Enactivism builds on situated and embodied cognition by arguing that cognition in biological systems is not only grounded in action (Noë 2004) but is also driven by a purpose to maintain its existence as a unity (Varela, Thompson, and Rosch 1991).

AI researchers have typically adopted approaches inspired by situated and embodied cognition either to exploit the advantages inherent in embodied agents (Brooks 1991), or because embodiment is considered a necessary condition of any model of animal or human cognition (Ziemke 2004). Attempts to apply situated and embodied AI in creative systems have often focussed on advantages to be gained by developing robotic systems but also tackle questions of grounding of creativity in artificial systems.

The marimba playing robot *Shimon* (`shimonrobot.com`) is an interesting example of an embodied creative system. Internally, the medium of music is represented as a choreography of gestures, opening up new opportunities for expressive performance (Hoffman and Weinberg 2010). Another example of an embodied creative system is the painting robot *e-David* (`e-david.org`), which attempts to approximate a given photograph through an iterative process of refinement; planning, applying then reviewing the results of paint strokes (Lindemeier et al. 2015), producing an ongoing 'conversation with the medium' (Schön 1983). Such embodied creative agents avoid symbolic representations by using "the world as its own model" (Brooks 1991) and engaging in a process similar to distributed cognition (Clark 1996). The performance of these creative systems, acting and responding to their physical and social environments, are reminiscent of craftsmen in action, potentially supporting the perception that the products of the machine labour is more authentic than that of a disembodied creative system.

Intrinsic Motivation Computational models of intrinsic motivation (Oudeyer 2008) allow developers of creative systems to further distance themselves; rather than provide externally defined goals, e.g., produce works in a given style, intrinsically motivated systems are provided with inherent drives, e.g., reward signals for the discovery of novelty (Schmidhuber 1991), maximising "empowerment" (Guckelsberger, Salge, and Colton 2017), or "learning progress" (Merrick and Maher 2009). Researchers in developmental robotics (Oudeyer 2012) use computational models of intrinsic motivation to produce embodied agents able to learn how to interact with their environment (Oudeyer, Kaplan, and Hafner 2007), in line with Turing's original vision.

Forms of intrinsic motivation explored in the development of creative systems include *curiosity*, the drive to discover novelty, and *competence*, the drive to master a skill. As an example of an embodied creative system, Merrick (2008) developed intrinsically motivated robot toys as a platform for stimulating creative play. Saunders, Chee, and Gemeinboeck (2013) developed a collective of curious robots that used their embodiment to reduce computational requirements. The process of training intrinsically motivated systems has similarities to an apprenticeship such that the creative system is guided through a sequence of learning experiences. What is not clear is how knowledge of such an apprenticeship might change the perception of authenticity.

Enactive Computational Creativity

The *enactive AI* framework (Froese and Ziemke 2009) adopts *autopoietic enactivism*, which roots intentional agency in the need of living organisms to self-produce through their ability to perceive and interact with their environment. Consequently, enactive AI extends embodied AI by grounding sensorimotor interaction in an agent's maintenance of its identity. Guckelsberger, Salge, and Colton (2017) argue that situated and embodied AI approaches do not go far enough in grounding creative systems but that *enactive AI* provides a framework for developing autonomous creative systems. From the perspective of developing autonomous creative systems, proponents of an enactive approach argue that simply being embodied is insufficient because it does not preclude the external assignment of values. For example, many of Shimon's goals are hard-coded, allowing it to improvise effectively with other musicians, while at the same time undermining its claim to autonomy because it does not act in these ways for its own purpose. Guckelsberger, Salge, and Colton (2017) propose that an enactive approach to the development of autonomous creative systems provides a method for escaping the imposition of human-given or hard-coded value systems, while at the same time conceding that such systems may not be recognised as creative due to the *embodiment distance* between the enactive system and the human observer.

Bridging the Embodiment Distance Given the challenge of adopting an enactive approach to creative systems, how might we bridge the embodiment distance between autonomous creative systems and human observers? A possible way forward is to recognise that the situation of an autonomous creative system includes the social and cultural environment, to the extent that it can share it. The embodied creative system "Curious Whispers" allows participants to interact using a three-button synthesiser, permitting the composition and performance of tunes similar to those shared between a group of robots (Saunders, Chee, and Gemeinboeck 2013). Opening up the collective to external perturbations in this way allowed some participants to inject cultural knowledge, in the form of simple tunes, into the collective memory of the agents through repeated performances. While not an enactive system, "Curious Whispers" suggests that by carefully designing creative systems to be open to their social and cultural environment in ways that allow the system to ground the incoming signals is one way that the distance between artificial and human embodiments may be bridged. Importantly, this process may open up autonomous creative systems to the social norms and cultural traditions that inform perceptions of authenticity.

Conclusions and Future Work

For culturally acceptable, truly autonomous creative behaviour in artificial systems, we believe a lack of authenticity is a looming issue. We have motivated and expanded on this belief here, and situated it in the context of *acceptable non-authenticity*, problems with a *lack of experience*, and notions of *expressive and brand authenticity*. As the quality of outputs increases, we can envisage an *uncanny valley* stretching out, where audiences marvel at the value of the products from creative systems, while despairing at the lack of authenticity in the process and in the nature of the originator. We have suggested software owning its non-authenticity, emphasising the product, producing speculative rather than realistic works of fiction, and building a brand as short-term ways in which to sidestep issues of authenticity. We have further suggested that software can record and later refer to its life experiences as a practical way in which to attain authenticity. Finally, we have discussed embodied Computational Creativity practices and proposals for enactive, purposeful computational creativity systems as ways in which we can engineer software which is simultaneously distanced from its programmers while closer to its audiences, potentially occupying an authentic position as an individual.

Addressing the (in)authenticity of software will contribute to the development of more sophisticated evaluation methods for Computational Creativity. These question the autonomy of the software, how it was constructed, what it does, how audiences and other stakeholders perceive it, how it presents its work through framing and other methods, and – as per the discussion here – should now also ask whether the software is seen as authentic in a particular project. As with these previous steps forward, we hope that acknowledging issues of inauthenticity will drive forward practical matters of engineering and deploying creative software, whether this involves simply avoiding asking software to autonomously generate emotion-laden poems about love, or developing embodied systems capable of grounding their 'life experiences' in order to authentically utilise them in future creative processes.

We hope to further highlight and unpick issues of authenticity in autonomously creative systems via the lens of existing theories on creative behaviour, such as the *Four Ps* breakdown of creativity into perspectives of person, process, product and press (Jordanous 2016). The discussion here is incomplete and too short to do justice to such a complex notion as computational authenticity, but we hope it provides a starting point for a conversation about what we believe will become an essential issue in Computational Creativity.

Acknowledgements

We wish to thank Christian Guckelsberger for his excellent and extensive comments on a draft of this paper, and the anonymous reviewers for many insightful suggestions. This work has been funded by EC FP7 grant 621403 (GRO).

References

Anderson, M. L. 2003. Embodied cognition: A field guide. *Artificial Intelligence* 149(1).

Baker, N. 2009. The Greatest Liar.
https://archives.cjr.org/second_read/the_greatest_liar_1.php

Barthes, R. 1967. The death of the author. *Aspen* 5-6.

Benjamin, W. 1968. The work of art in the age of mechanical reproduction. In *Illuminations*. Shocken Books.

Brooks, R. A. 1991. Intelligence without reason. In *Proceedings of the 12th International Joint Conference on AI*.

Clancey, W. J. 1997. *Situated Cognition: On Human Knowledge and Computer Representations*. CUP.

Clark, A. 1996. *Being There: Putting Brain, Body, and World Together Again*. MIT Press.

Cohen, E. 1988. Authenticity and commoditization in tourism. *Annals of Tourism Research* 15(3).

Colton, S.; Cook, M.; Hepworth, R.; and Pease, A. 2014. On acid drops and teardrops: Observer issues in Computational Creativity. In *Proceedings of the AISB symposium on AI and Philosophy*.

Colton, S. 2012. The Painting Fool: Stories from building an automated painter. In McCormack and d'Inverno (eds) *Computers and Creativity*. Springer.

Dutton, D. 2003. Authenticity in art. In Levinson, J., ed., *The Oxford Handbook of Aesthetics*. OUP.

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 1706.07068*.

Errington, P. W. 2017. *J.K. Rowling: A Bibliography*. Bloomsbury.

Frey, J. 2003. *A Million Little Pieces*. Random House.

Frey, J. 2006. A note to the reader.
penguinrandomhouse.biz/media/pdfs/AMLP020106.pdf

Froese, T., and Ziemke, T. 2009. Enactive artificial intelligence: Investigating the systemic organization of life and mind. *Artificial Intelligence* 173(3).

Galbraith, R. 2013. *The Cuckoo's Calling*. Sphere.

Gatys, L.; Ecker, A.; and Bethge, M. 2016. Image style transfer using convolutional neural networks. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*.

Glaveanu, V. 2017. Creativity in craft. In *The Cambridge Handbook of Creativity Across Domains*. Cambridge University Press.

Guckelsberger, C.; Salge, C.; and Colton, S. 2017. Addressing the "why?" in Computational Creativity: A non-anthropocentric, minimal model of intentional creative agency. In *Proc. 8th ICCC*.

Hoffman, G., and Weinberg, G. 2010. Gesture-based human-robot jazz improvisation. *IEEE Int. Conf. on Robotics & Automation*.

Jacobson, D. 2018. RockNotes: Punks vs. Poseurs. beachwoodreporter.com/music/rocknotes_punks_vs_poseurs.php

Jordanous, A. 2016. Four PPPPerspectives on Computational Creativity in theory and in practice. *Connection Science* 28.

Lakoff, G., and Johnson, M. 1999. *Philosophy in the Flesh: The Embodied Mind and Its Challenge to Western Thought*. Collection of Jamie and Michael Kassler. Basic Books.

Lindemeier, T.; Metzner, J.; Pollak, L.; and Deussen, O. 2015. Hardware-based non-photorealistic rendering using a painting robot. In *Computer Graphics Forum* 34.

Lindholm, C. 2013. The rise of expressive authenticity. *Anthropological Quarterly* 86(2).

Llano, M.; Colton, S.; Hepworth, R.; and Gow, J. 2016. Automated fictional ideation via knowledge base manipulation. *Journal of Cognitive Computation* 8(2).

MacCannell, D. 1973. Staged authenticity: Arrangements of social space in tourist settings. *The American J. of Sociology* 79(3).

Macedo, L., and Cardosa, A. 2001. Creativity and surprise. In *Proc. AISB Symp. on AI and Creativity in Arts and Science*.

Merrick, K. 2008. Designing toys that come alive: Curious robots for creative play. In *Proc. Seventh International Conference on Entertainment Computing*. Springer.

Merrick, K. E. and Maher, M. L. 2009. *Motivated Reinforcement Learning*. Springer.

Morhart, F.; Malär, L.; Guèvremont, A.; Girardin, F.; and Grohmann, B. 2015. Brand authenticity: An integrative framework and measurement scale. *J. of Consumer Psychology* 25.

Noë, A. 2004. *Action in Perception*. MIT Press.

Oudeyer, P.-Y.; Kaplan, F.; and Hafner, V. V. 2007. Intrinsic motivation systems for autonomous mental development. *IEEE Transactions on Evolutionary Computation* 11(2).

Oudeyer, P.-Y. 2008. How can we define intrinsic motivation? In *Proc. 8th Int. Conf. on Epigenetic Robotics: Modeling Cognitive Development in Robotic Systems*. 93–101

Oudeyer, P.-Y. 2012. Developmental robotics. In Seel, N. M., ed., *Encyclopedia of the Sciences of Learning*. Springer, 969–972

Pease, A., and Colton, S. 2012. On impact and evaluation in Computational Creativity: A discussion of the Turing test and an alternative proposal. In *Proc. of the AISB symp. on AI and Philosophy*.

Pérez y Pérez, R. 2017. Mexica: 20 years – 20 stories. Counterpath.

Saunders, R., and Gero, J. S. 2001. A curious design agent: A computational model of novelty-seeking behaviour in design. In *Proceedings of CAADRIA 2001*.

Saunders, R.; Chee, E.; and Gemeinboeck, P. 2013. Evaluating human-robot interaction with embodied creative systems. In *Proceedings of the 4th Int. Conference on Computational Creativity*.

Schmidhuber, J. 1991. A possibility for implementing curiosity and boredom in model-building neural controllers. In *Proc. of the Int. Conf. on Simulation of Adaptive Behaviour*.

Schön, D. A. 1983. *The reflective practitioner: How professionals think in action*. Basic Books.

Seyama, J., and Nagayama, R. S. 2007. The uncanny valley: Effect of realism on the impression of artificial human faces. *Presence: Teleoper. Virtual Environ.* 16(4).

Slater, L. 2004. *Opening Skinner's Box: Great Psychology Experiments of the Twentieth Century*. W. W. Norton and Company.

Turing, A. 1948. Intelligent machinery. Technical report, NPL.

Turing, A. 1950. Computing machinery & intelligence. *Mind* 49.

Varela, F. J.; Thompson, E.; and Rosch, E. 1991. *The Embodied Mind: Cognitive Science and Human Experience*. MIT Press.

Varga, S. 2013. *Authenticity as an Ethical Ideal*. Routledge.

Wang, N. 1999. Rethinking authenticity in tourism experience. *Annals of Tourism Research* 26(2).

Wimsatt, W., and Beardsley, M. 1954. *The Verbal Icon: Studies in the Meaning of Poetry*. University of Kentucky Press.

Wood, A. M.; Linley, P. A.; Maltby, J.; Baliousis, M.; and Joseph, S. 2008. The authentic personality: A theoretical and empirical conceptualization and the development of the authenticity scale. *Journal of Counseling Psychology* 55(3).

Wyatt, E. 2006. Author Is Kicked Out of Oprah Winfrey's Book Club. nytimes.com/2006/01/27/books/27oprah.html

Ziemke, T. 2004. Embodied AI as science: Models of embodied cognition, embodied models of cognition, or both? *Embodied Artificial Intelligence*. Springer LNCS 3139.

Performative Body Mapping for Designing Expressive Robots

Rob Saunders[1,3] and **Petra Gemeinboeck**[2,3]
[1]Design Lab, Sydney University, Australia
[2]Creative Robotics Lab, University of NSW, Australia
[3]The MetaMakers Institute, Falmouth University, UK

Abstract

This paper explores the challenges and opportunities of skill acquisition for creative robotics, where the required knowledge is highly embodied. We present Performative Body Mapping (PBM) as a suitable methodology for harnessing the movement expertise of trained professionals. We describe the results from a series of workshops to design and train a non-humanlike robot through embodied explorations of possible forms and movements. In addition to the PBM methodology, we propose a method for evaluating expressive robot performers by adapting the Godspeed questionnaire, commonly used in social robotics, which gathers audience feedback on the perception of five properties of interest in creative robotics; anthropomorphism, affective agency, intelligibility, perceived intelligence, and perceived originality. We report on some preliminary results from a first audience study of an early prototype of our robot and discuss the implications for our research.

Introduction

The field of creative robotics lies at the intersection of computational creativity and social robotics, it is concerned with both the development of embodied creative systems and the application of creative practices to further human-robot interaction (Koh et al. 2016; Gemeinboeck 2017). The project described here straddles these approaches by exploring the role that movement experts, e.g., dancers and choreographers, can play in the design and training of non-anthropomorphic robots and the ability for trained robots to improvise novel movements. Using the design of a non-anthropomorphic robot as a platform, we address questions of skill acquisition across different embodiments, i.e., human and robotic, in a domain where knowledge is tacit, unstructured and resistant to formalising due to its embodied nature (Csikszentmihalyi 1988). Our focus in this paper is on the capture and reproduction of improvised movements from experts, the engagement of an audience through movement, and the perception of agency when a robot performs.

Embodied, Enactive and Distributed Creativity

Computational creativity, like many other scientific fields of creativity research, has tended to emphasise the thinking over making, i.e., ideation over the craft-like activities that support creativity (Glăveanu 2017). Unsurprisingly for a subfield of AI, computational creativity draws extensively on representational theories of creativity from cognitive science, e.g., the highly influential work of Boden (1990; 1994a; 1994b). Malafouris argues, however, that representational theories of creativity, like those of Boden, tend "to reduce the rich ecology of the creative space to some internalised 'problem space' that can be mentally manipulated and transformed to produce some creative result" (Malafouris 2014, p.145). Where the 'rich ecology of the creative space' that Malafouris laments is composed of the material, technical, social and cultural milieu that human creativity both exists within and continuously transforms.

Theories of embodied, enacted, and distributed cognition provide alternative perspectives on notions of creativity (Varela, Thompson, and Rosch 1991; Lakoff and Johnson 1999; Clark 1996; Noë 2004). Some have begun to explore approaches to computational creativity based on enactive cognition, e.g., see Takala (2015). Guckelsberger, Salge, and Colton (2017) argue that *enactive AI* (Froese and Ziemke 2009) provides the most suitable framework for developing autonomous creative systems, while conceding that such systems may not be recognised as creative due to the *embodiment distance* between computational system and human audiences. The challenge of bridging the embodiment distance is significant but building computational systems that are grounded in, and can engage with, the ecology that human creativity constructs and relies upon, may be key.

The overarching aim of the our project is to explore pragmatic methods for producing situated, embodied actors that balance the needs of grounding and evolving creative skills based on its (1) material, social and cultural situation, and (2) machinic embodiment. Our approach relies on working closely with experts, who provide the material, social and cultural situation that inform the design and training of a robot. Contemporary dance deliberately and systematically cultivates movement for its own sake (Stevens and McKechnie 2005, p.243), making it an ideal domain of expertise to draw upon, especially, given its practitioners willingness to engage with questions related to the bridging of human and non-human forms of embodiments through movement.

This paper presents a methodology for the design of creative robotics that focuses on the analysis and design of movements based on the kinaesthetic expertise of choreographers and dancers. We begin by exploring the perception of agency based on the movement of non-anthropomorphic robots within a specific social context by examining notions of agency in robotic art and performance. We look at the challenges faced in robotics of producing movements that convey affect, in particular the *correspondence problem*, i.e., the mapping of movement between humans and robots with different embodiments. We propose a methodology, called Performance Body Mapping (PBM), as an approach for bridging between different embodiments by leveraging the ability of movement experts, e.g., dancers, to inhabit and animate non-human forms. To study the capacity for our cube-like robot to elicit affect, we have developed an instrument for conducting audience surveys, based on the Godspeed questionnaire, widely used in social robotics. We report on a first audience study of an early prototype of a cube-like robot, and discuss implications for future work.

Background

Researchers in social robotics often rely on an underlying assumption that anthropomorphic or zoomorphic appearance assists the formation of meaningful connection between humans and robots (Duffy 2003). A number of projects have explored different machine learning methods for teaching humanoid robots how to move based on the recording of humans dancing (Ros, Baroni, and Demiris 2014; Özen, Tükel, and Dimirovski 2017) and recently the creation of novel dances for humanoid robots based on motion capture data has been explored (Augello et al. 2016; 2017), as well as, the potential for co-creativity with humanoid robots (Fitzgerald, Goel, and Thomaz 2017).

Building robots in our own image, however, deliberately blurs the distinction between organic and mechanical bodies, cognition and computation, to elicit human investment based on superficial and often false social cues. Studies in Human-Robot Interaction (HRI) illustrate the difficulty of the underlying assumption, by highlighting the frustration and disappointment experienced by humans when the social capabilities of a robot fall short of their expectations based on its appearance (Dautenhahn 2013). Non-anthropomorphic robots, on the other hand, permit human–machine encounters that aren't restricted by "preconceptions, expectations or anthropomorphic projections...before any interactions have occurred" (Dautenhahn 2013).

The Perception of Animacy

The challenge for developing non-anthropomorphic robots for performance is to design the affective potential that is inherent in movement such as to elicit desired responses in the viewer. The potential for simple movements of geometric shapes to be perceived to indicate high-level properties such as causality and animacy has been studied for over 80 years (Scholl and Tremoulet 2000). The phenomenon first documented by Michotte (1963) and Heider and Simmel (1944) is often illustrated with Michotte's "launching effect" where one small object (A) moves until it is adjacent to another item (B), at which point A stops and B starts moving. Different spatial and temporal relationships between the movements of A and B, result in different causal relations being perceived by viewers, regardless of cultural background.

Beyond causality, the principle of animacy also appears to be perceived in the movements of abstract shapes. Studies of perceptual animacy typically involve the perception of a simple shape being alive and often gives rise to the perception of goals, e.g., 'trying to get over here', or mental states, e.g., 'wanting to get over there' (Heider and Simmel 1944). Recently, building on theories of the perception of animacy, Levillain and Zibetti (2017) proposed a theoretical framework for understanding the agency ascribed to 'behavioural objects', such as robotic artworks.

Artists have long understood the power of movement in non-anthropomorphic machines to elicit audience responses. For example, Simon Penny created *Petit Mal*, an autonomous wheeled robot that would interact with gallery visitors, to produce 'behaviour which was neither anthropomorphic nor zoomorphic, but which was unique to its physical and electronic nature' (Penny 2000). Experiments with choreographing robots can be traced back to 1973 and the pioneering work of Margo Apostolos (Apostolos 1990). A number of choreographers have experimented with robots in their performances since, including Pablo Ventura, Thomas Freudlich and Huang Yi. In many of these works, the movement expertise of the choreographer transforms non-anthropomorphic robots into expressive bodies that can be read by human audiences. They have relied, however, on the ability of the choreographer to program a robot to reproduce movements exactly as instructed.

The Correspondence Problem

In HRI, a range of methods have been developed to specify a robot's movements, from a programmer "imagining a movement executed by the robot's body" to produce a sequence of instructions (Alac 2009), to *programming by demonstration* (Billard et al. 2008) where the movements of a human are captured for a robot to learn to imitate. The former is challenging because it requires the programmer to translate the (imagined) movement into a precise algorithmic representation. The challenge of the latter approach is the translation between different physical embodiments, known as the *correspondence problem*, i.e., the problem of mapping between a human body and a robot with a different morphology, movement repertoire and sensorimotor capabilities (Dautenhahn, Nehaniv, and Alissandrakis 2003).

To overcome the correspondence problem, researchers construct complex mappings between the movements of a human and the corresponding movements of a robot. In non-anthropomorphic robots this is particularly challenging and often results in engineers making a series of assumptions about the mapping that may or may not be informed by expertise in movement. Despite this challenge, programming by demonstration or *demonstration learning* is a popular approach, because it makes it possible for robots to learn behaviours and skills without every action they perform needing to be explicitly and painstakingly programmed (Dautenhahn, Nehaniv, and Alissandrakis 2003). The following section discusses Performative Body Mapping (PBM), which builds on the core ideas of demonstration learning but delegates much of the difficult morphological mapping to movement experts (Gemeinboeck and Saunders 2014).

Methodology

Performative Body Mapping has been developed to harness the ability of performers to map between different body morphologies. It is comprised of four stages; *bodying*, *grounding*, *imitation*, and *improvisation*. Here we focus on the first stage, which includes form finding, motion capture and the prototype construction, for more information on the complete process see Gemeinboeck and Saunders (2014). Bodying is concerned with the design of a robot's form in tandem with its movement capabilities. Often the design of a robot's physical form is dominated by functional requirements that manifest humanistic assumptions about the ways a robot can or should move (Ziemke 2016). Even in social robotics, where interaction with humans is paramount, movement is often a secondary concern to appearance. In contrast, the PBM requires that form and movement be designed in concert using an iterative approach.

Designing through Movement

To iteratively 'find' and refine the robot's form, PBM involves the use of a wearable object, or 'costume', resembling a possible robot form, that can be inhabited and animated by a dancer. Costumes have been used by choreographers and dramaturgists to co-shape and transform dancers' movements, see Schlemmer's Bauhaustänze in Birringer (2013) and Heiner Müller's Tristan and Isolde in Suschke (2003). In PBM, involving the bodily imagination (DeLahunta, Clarke, and Barnard 2012) and kinesthetic empathy (Reynolds 2012) of dancers, allows the costume to become an efficient instrument for mapping between very different embodiments. In particular, the use of a costume; (1) provides dancers with an embodied insight into the material and morphological characteristics of a robot, (2) supports the development of a repertoire of movements and movement qualities, and (3) allows movement data to be captured that a robot can learn from, with little or no translation. The shape of the costume was not fixed during this stage and was redesigned in response to the movements and bodily relations the dancers could activate. The dancer's movements, in turn, were co-shaped by the affordances of the costume, so that distinct movement qualities could emerge from a material interdependence between the two.

We collaborated with the De Quincey Company[1] and its artistic director and choreographer Tess de Quincey. The De Quincey Company practice *BodyWeather*, which draws from both Eastern and Western dance traditions, sports training, martial arts and theatre practices. BodyWeather practitioners are well attuned to the task of bodily thinking through 'other' body-forms, in Tess de Quincey's words, "the whole point about BodyWeather is to go beyond the biomechanics through images, [that is] we recruit the biomechanics to find ways to move, which are not normally positioned as human movements" (De Quincey 2015).

During the early movement studies the dancers inhabited a wide range of materials and objects to narrow the scope of possible robot forms. Our goal was to find forms that foregrounded movement over appearance and avoided analogies with living 'things'. Enabling constraints for the exploration included that the form should be without a front or back, head or face, or limb-like structures, and that it should be technically possible to construct robot capable of imitating the costume's movements. This process quickly filtered out forms that, when activated, either relied too heavily on the dancer's body, would be impossible to construct, or were perceived as relying too heavily on its novel appearance.

(a) Cardboard box inhabited by Linda Luke.

(b) Costume with markers inhabited by Kirsten Packham.

(c) Robot motion testing.

(d) Robot as 'plinth'.

Figure 1: Evolution from costume (a,b) to prototype (c,d). Photos © Petra Gemeinboeck

As dancers experimented with geometric forms, it became apparent that the simpler the form, the more our focus shifted towards the movement of the costume. Ultimately, this lead to using the most obvious abstract form, yet not the most apparent in terms of its evocative capacity—a box. The dancers started by inhabiting a 150x55x45cm cardboard box, see Figure 1a. Iterations on the design reduced the height of the box until it became a cube, further distancing it from human proportions and focussing attention on the movement. The dancers noted that the box became particularly interesting when it balanced precariously on an edge or was tipped onto one corner. Confronting our notions of weight and gravity through tilting, swaying and teetering allowed for the box to lose its stability and, with it, its 'boxiness'. The ability to reproduce these types of movements became a primary goal for the design of the robot prototype.

Motion Capture and Machine Learning

The motion of the activated costume was tracked to (1) inform the model for a mechanical prototype that resembles the costume and its capacities to move as closely as possible, and (2) provide data for the robot to learn from. The cube's

[1] http://dequinceyco.net

movements were captured using a video-based motion tracking system by attaching coloured targets to the cube's surfaces, as can be seen in Figure 1b. Activated by a dancer, the movements of the cube were recorded using two HD cameras arranged to ensure that all sides of the cube, except the base, could be seen at all times. The video recordings were analysed using custom motion tracking software and the resulting tracked 3D points were used to extract the cube's position (x, y, z) and the orientation (yaw, pitch, roll).

In total, we captured approx. 15 hours of movement data from three dancers over a period of five days. From this dataset we initially extracted five hours of motion capture data that represented the types of movement sequences that we wanted to test in the Re/Pair exhibition, see Results. To reduce ambiguity in the interpretation of the captured data, an inverse kinematic model of the robot was developed based on two joints, one to represent the (x, y, z) position of the base of the robot and one to represent the (yaw, pitch, roll) orientation of the top, relative to the base. The motion capture data was processed using the inverse kinematic model to derive the position and orientation of the two joints, the resulting data set consisted of 360,000 joint positions.

We applied a mixture density LSTM network, previously used to successfully synthesise handwriting (Graves 2014) and choreography (Crnkovic-Friis and Crnkovic-Friis 2016). The inputs and outputs of the neural network were 6-dimensional tensors (x, y, z, yaw, pitch, roll) and the architecture consisted of 3 hidden layers of 512 neurons, a total of approx. 5.3M weights. The synthesised movement sequences were subjectively assessed by experts against the original performances of the dancers before adding them to a catalogue of possible movement sequences. In addition to expanding the repertoire of movements, the aim at this stage was to produce a baseline result for future comparisons with the 'grounded' approach outlined later, see Discussion.

Robot Prototype

The video and motion capture data were analysed to determine the degrees of freedom required to replicate the movements of the dancers. To achieve these requirements, the design of the robot combines two main subcomponents; (1) a 'Kiwi Drive'—an omni-directional wheeled base (Pin and Killough 1994) that provides 3 degrees of freedom (x, y, yaw), and (2) a 'Stewart Platform' (Stewart 1965) that provides 6 degrees of freedom relative to the base (x, y, z, yaw, pitch, roll). The former allows the robot to turn on the spot and move over the ground plane without first having to turn to face the direction of travel. The latter allows the robot to shift, tilt and rotate by smaller amounts, relative to the base.

The use of omnidirectional wheels ensures that the robot design maintains an important initial criteria of the movement studies because the resulting robot has no front or back, a necessary condition for replicating some of the movement sequences recorded where the dancer quickly changed the direction of travel. The Stewart platform provides the flexibility necessary to reproduce the range of angles recorded for pan, tilt and yaw as well as the speed to produce some of the smaller, sudden or subtle movements produced by the dancers. Figure 1c is a photo of the robot prototype without its outer cover being tested for range of movement.

The robot prototype was shown in the Re/Pair exhibition, part of the Big Anxiety Festival[2] at the University of New South Wales, Sydney, Australia. The Re/Pair exhibition brought together five robotic works in different stages of development. Figure 1d shows the completed robot with its outer cover that was designed to mimic the plinths used in the gallery setting, while also maintaining the shape of the original costume. The main goal for exhibiting the robot at this early stage was to survey audience members regarding their perception of the robot's agency and originality.

Evaluation of Affective Agency

Several methods have been used to evaluate the perceptions and impressions of social robots (Walters et al. 2013; Vlachos and Scharfe 2015). The Godspeed Questionnaire Series (GQS) is one of the most frequently used and influential tools for evaluating social robots (Bartneck, Croft, and Kulic 2009). The GQS addresses five key concepts: *Anthropomorphism*, *Animacy*, *Likeability*, *Perceived Intelligence*, and *Perceived Safety*. These concepts are particularly significant for social robotics where the safety and the ability of users to relate to a robot are of paramount concern.

For a performance based on the movement expertise of contemporary dancers our primary concern is whether the "[m]ovement material that is created, performed, or observed engages motor and kinaesthetic processes and leads to cognitive and affective reactions" (Stevens and McKechnie 2005, p.1570). Consequently, we developed a questionnaire based on the GQS to address key concepts more appropriate to the evaluation of our research questions, i.e., *Anthropomorphism*, *Affective Agency*, *Intelligibility*, *Perceived Intelligence* and *Perceived Originality*. The choice of these concepts was driven by our desire to understand how the movement of the robot prototype is perceived in terms of affect, and how this perception is related to the perception of anthropomorphic qualities. The other perceptions we were interested in relate to computational creativity, such as, the perceived intelligence and originality, as well as, the intelligibility of the robot's movements.

To confirm the internal consistency and the validity of our data, an internal reliability test was conducted. The results showed that the *Anthropomorphism* and *Affective Agency* indices had the highest reliability with a Cronbach's alpha of 0.84 and 0.82 respectively, followed by *Intelligibility* and *Perceived Intelligence* indices with a Cronbach's alpha of 0.75 and 0.74 respectively, and *Perceived Originality* had a Cronbach's alpha of 0.70, meeting the standard 0.70 threshold (Nunnally 1978).

Results

During the Re/Pair exhibition we collected a total of 48 questionnaires. The majority of the participants were between 21 and 55 years old. As with other "in the wild" experiments, context plays an important role in evaluation, consequently we sought to maintain the gallery context until

[2]https://www.thebiganxiety.org/events/repair/

participants were asked to fill in a questionnaire. Participants were not made aware of the research components of this study ahead of time, rather, they were asked to fill out a questionnaire only after being observed interacting with the robot. The majority of the participants (81%) reported that they engaged with the robot for more than 2 minutes, and half the participants reported that they engaged with the robot for more than 5 minutes.

Participants were given a list of possible reasons for what attracted them to engage with the robot in the gallery, from which they could choose multiple items. The responses were grouped into 5 categories: the sound of the robot, the appearance of the robot, the movement of the robot, the project description, and other. 36 (75%) of the participants responded that the robot's movement attracted them, 23 (48%) reported that the project description drew their attention, while 17 (35%) cited the appearance of the robot and 5 (10%) the sound the robot made. 10 (21%) of the participants gave other reasons for being attracted to the robot.

Figure 2 illustrates the questionnaire responses as box plots of the participants' ratings for each of the five indices, using the Tukey convention with the median values and the box indicating the first and third quartiles, the whiskers indicate the lowest and highest datum within 1.5 IQR (interquartile range) of the lower and upper quartile, outliers are indicated with crosses (Tukey 1977). Detailed analysis of the results of the questionnaire is depicted in Table 1 and shows that the robot received high ratings for *Affective Agency* ($M = 3.43$), moderately high ratings for *Perceived Intelligence* ($M = 3.06$) and *Perceived Originality* ($M = 2.95$), moderately low ratings for *Anthropomorphism* ($M = 2.02$), and varied responses for *Intelligibility* ($M = 2.56, SD = 1.21$).

Discussion

The goal of our first evaluation was to examine whether the 'bodying' stage of the PBM methodology would permit movement experts to design and train a non-humanlike robot to perform in ways that are expressive and engaging. The results indicate that the primary reason for people to engage with the robot was movement ($n = 36$), significantly more than appearance ($n = 17$), while audience members were clear that the robot was non-anthropomorphic ($M = 2.02$). While this is not surprising, given the simple appearance of the robot and the environment it was placed within, it suggests that movements like those performed by the robot can be a significant attractor, without the need for an overtly anthropomorphic appearance. This aligns with Levillain and Zibetti's observations of the attraction of robotic artworks as 'behavioural objects' (2017), although it may also suggest an attraction to the novelty of the object given that, despite the robot being unable to create significantly novel movements, participants rated the perceived originality of the robot relatively high ($M = 2.95$). The effect of novelty is something that we will investigate further in future studies when we explore how the ability to improvise novel movements affects audience perception.

Participants rated the ability of the robot to produce affect highly ($M = 3.43$), suggesting that the robot was able to sufficiently reproduce some of the qualities of the dancers' movements to elicit an affective response. They also perceived the robot to have higher intelligence than we might have expected ($M = 3.06$) given that this early prototype could not interact with visitors, this may have been a consequence of the unexpected complexity and nuance of the movements. We observed, however, that visitors often adapted their own behaviour to accommodate the robot and this may explain a higher than expected perception of intelligence. In future studies we will explore how the perception of intelligence is affected as we add the ability for it to sense its environment through the addition of sensors.

Future Work

This study involved an early robot prototype and has investigated only the first stage of PBM, i.e., bodying. The ability of the robot to engage gallery visitors through movement and the audience perceptions of affective agency and intelligence suggest that, even at this early stage, PBM supports the ability of movement experts to embody a non-anthropomorphic form and map from their embodiment to that of the robot. The remaining stages in the PBM methodology are concerned with the *grounding* of the robot's movement, *imitation* through sequence learning, and *improvisation* using intrinsically motivated learning.

The motor controller used in the robot prototype decomposes the problem along functional lines between the Kiwi Drive and the Stewart Platform. The *grounding* stage will use 'active motor babbling' (Saegusa et al. 2009; Baranes and Oudeyer 2013) to derive a controller that bidirectionally maps between the motor and sensor data of the robot. The resulting forward and inverse mappings will provide a richer model for the application of sequence learning (Graves 2014; Crnkovic-Friis and Crnkovic-Friis 2016) in the *imitation* stage to take advantage of redundancy in the movement capabilities of the two subcomponents, i.e., for small movements in x, y and yaw, and the spatiotemporal context within movement sequences, e.g., to anticipate future movements. Finally, the *improvisation* stage will use intrinsically motivated learning (Baranes and Oudeyer 2010) to expand the repertoire of movements, beyond the generalisations produced by the *imitation* stage based on the grounded sensorimotor mapping.

The movement centric approach to the design and training of a non-anthropomorphic robot, which is at the core of PBM, provides another method for tackling the *correspondence problem* frequently encountered in demonstration learning. We will continue to apply PBM to robots' performance in theatrical, artistic and social situations but future applications of PBM could include the acquisition of other embodied skills that support creative activity across a range of domains, e.g., traditional crafts.

Keith Sawyer distinguishes between the study of 'product creativity' and 'performance creativity'; where the former studies what remains after the creative act, e.g., scores, paintings, sculptures, while in the latter "the creative process and the resulting product are co-occurring" (Sawyer 1998, p.11). Much of computational creativity, like psychology, has focussed on product creativity but Sawyer observes

Table 1: Analysis of Questionnaire Responses

Attribution	Attributes	Mean (M)	Standard Deviation (SD)
Anthropomorphism	Mechanical — Organic	2.13	1.18
α = 0.84	Machine-like — Human-like	2.07	1.16
M = 2.02	Non-human — Human	1.75	1.24
SD = 1.21	Artificial — Natural	1.79	1.23
	Machine — Performer	2.36	1.19
Affective Agency	Bland — Expressive	3.51	0.81
α = 0.82	Forgettable — Memorable	3.34	1.08
M = 3.43	Dull — Evocative	3.64	0.93
SD = 0.97	Trivial — Meaningful	3.03	0.99
	Boring — Engaging	3.63	0.94
Intelligibility	Unintelligible — Intelligible	3.14	0.96
α = 0.75	Enigmatic — Understandable	1.95	1.03
M = 2.56	Opaque — Readable	2.51	1.19
SD = 1.21	Ambiguous — Obvious	1.81	1.04
	Unconvincing — Believable	3.39	0.98
Perceived Intelligence	Incompetent — Competent	2.94	1.17
α = 0.74	Unintelligent — Intelligent	2.91	1.07
M = 3.06	Aimless — Deliberate	2.92	1.15
SD = 1.14	Indifferent — Curious	3.61	0.95
	Scripted — Imaginative	2.94	1.21
Perceived Originality	Simple — Puzzling	2.84	1.33
α = 0.70	Predictable — Surprising	3.31	1.10
M = 2.95	Scripted — Imaginative	2.94	1.21
SD = 1.22	Rehearsed — Spontaneous	3.09	1.20
	Rigid — Elastic	2.58	1.16

performance creativity “may actually represent a more common, more accessible form of creativity than privileged domains such as the arts and sciences” (Sawyer 1998, p.12). One of the challenges of this view for computational creativity is the development of creative systems capable of enacting a constructive dialogue with the world (Schön 1983). Performative, embodied approaches like PBM may provide a fruitful approach to the development of such systems. If we succeed it may tell us more about the lived experience of being creative than representational theories of creativity.

Conclusion

This paper has briefly made the case for creative systems, and creative robots in particular, to acquire embodied, craft-like skills as an alternative to following representational theories of creativity. A significant challenge in acquiring traditional embodied skills is the mapping between the embodiment of a human and that of a robot. We have proposed Performative Body Mapping as a methodology for the design and training of robots for the purpose of acquiring embodied skills. This paper has described the application of the ‘bodying’ stage of PBM to the design and training of a non-humanlike robot by movement experts for the purpose of performing in a gallery context. The audience survey suggests that this process of dancers inhabiting and animating abstract robot forms, successfully harnesses their embodied skills to design and train a non-humanlike robot with a capacity to be perceived as an affective agent.

Acknowledgments

This research is supported under the Australian Research Council’s Discovery Projects funding scheme (project number DP160104706).

References

Alac, M. 2009. Moving android: On social robots and body-in-interaction. *Social Studies of Science* 39(4):491–528.

Apostolos, M. K. 1990. Robot choreography: Moving in a new direction. *Leonardo* 23(1):25–29.

Augello, A.; Infantino, I.; Manfrè, A.; Pilato, G.; Vella, F.; and Chella, A. 2016. Creation and cognition for humanoid live dancing. *Robot. Auton. Syst.* 86:129–137.

Augello, A.; Cipolla, E.; Infantino, I.; Manfre, A.; Pilato, G.; and Vella, F. 2017. Creative robot dance with variational encoder. In Goel, A.; Jordanous, A.; and Pease, A., eds., *Proc. 8th Int. Conf. Computational Creativity*, 41–48.

Baranes, A., and Oudeyer, P.-Y. 2010. Intrinsically motivated goal exploration for active motor learning in robots: A case study. In *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS 2010).*

Baranes, A., and Oudeyer, P.-Y. 2013. Active learning of inverse models with intrinsically motivated goal exploration in robots. *Robot. Auton. Syst.* 61(1):49–73.

Bartneck, C.; Croft, E.; and Kulic, D. 2009. Measurement instruments for the anthropomorphism, animacy, likeability,

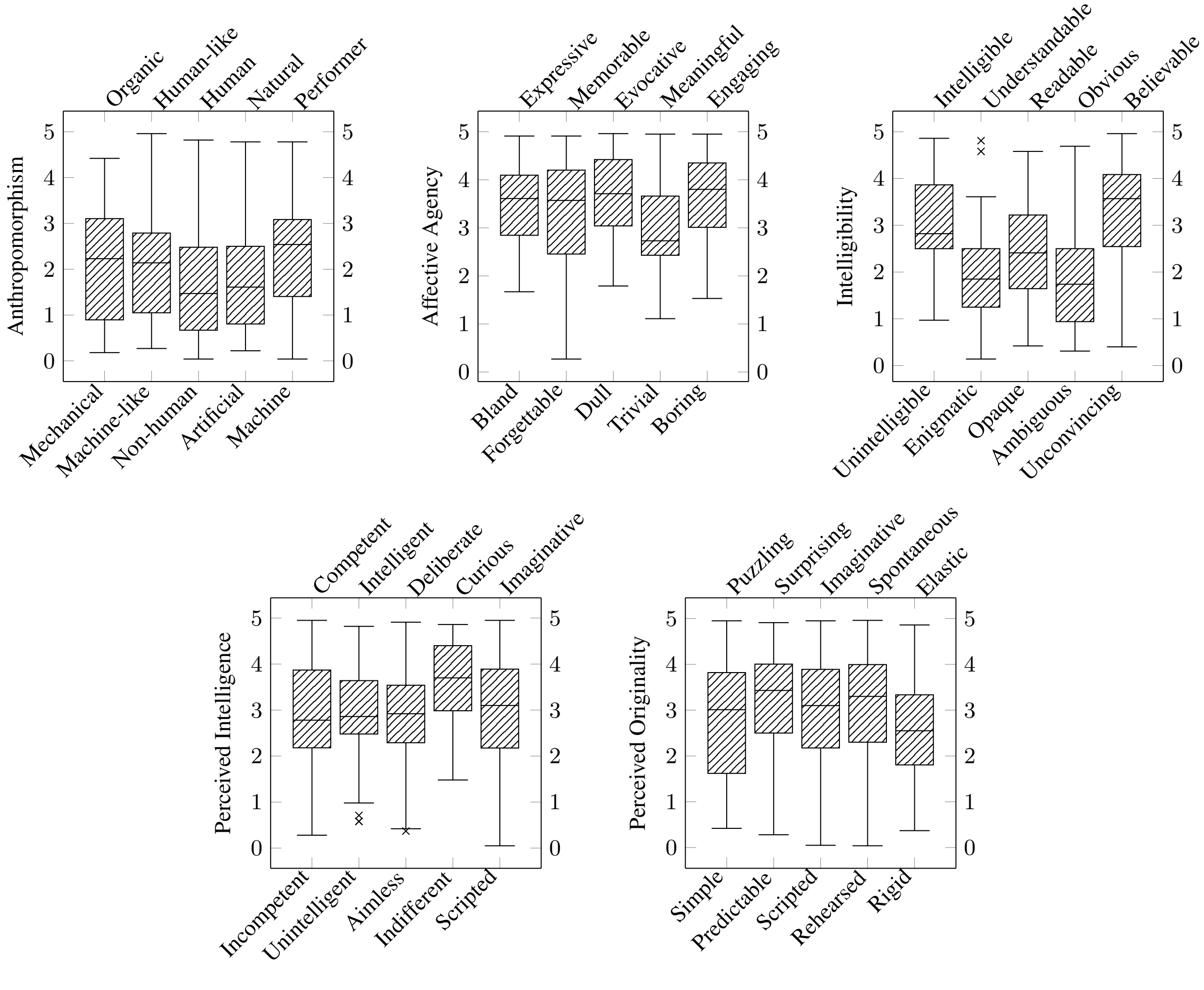


Figure 2: Analysis of Questionnaire Responses

perceived intelligence, and perceived safety of robots. *International Journal of Social Robotics* 1(1):71–81.

Billard, A.; Calinon, S.; Dillmann, R.; and Schaal, S. 2008. Robot programming by demonstration. In *Springer Handbook of Robotics*. Berlin: Springer. 1371–1394.

Birringer, J. 2013. Bauhaus, constructivism, performance. *PAJ: A Journal of Performance and Art* 35(2):39–52.

Boden, M. A. 1990. *The Creative Mind: Myths and Mechanisms*. London: Cardinal.

Boden, M. A. 1994a. *Dimensions of creativity*. Cambridge, MA: MIT Press.

Boden, M. A. 1994b. What is creativity? In Boden, M. A., ed., *Dimensions of Creativity*. Cambridge, MA: The MIT Press. 75–117.

Clark, A. 1996. *Being There: Putting Brain, Body, and World Together Again*. Cambridge, MA: MIT Press.

Crnkovic-Friis, L., and Crnkovic-Friis, L. 2016. Generative choreography using deep learning. In *Proc. 7th Int. Conf. Computational Creativity*, 272–277.

Csikszentmihalyi, M. 1988. Society, culture and person: a systems view of creativity. In Sternberg, R. J., ed., *The Nature of Creativity: Contemporary Psychological Perspectives*. Cambridge, UK: Cambridge University Press. 325–339.

Dautenhahn, K.; Nehaniv, C.; and Alissandrakis, A. 2003. Learning by experience from others—social learning and imitation in animals and robots. In Kühn, R.; Menzel, R.; Menzel, W.; Ratsch, U.; Richter, M.; and Stamatescu, I., eds., *Adaptivity and Learning: An Interdisciplinary Debate*. Berlin: Springer. 217–421.

Dautenhahn, K. 2013. Human–robot interaction. In Soegaard, M., and Dam, R. F., eds., *Encyclopedia of Human-Computer Interaction*. Aarhus: Interaction Design Foundation.

De Quincey, T. 2015. Video recording. unpublished.

DeLahunta, S.; Clarke, G.; and Barnard, P. 2012. A conversation about choreographic thinking tools. *Journal of Dance & Somatic Practices* 3(1–2):243–59.

Duffy, B. R. 2003. Anthropomorphism and the social robot. *Robot. Auton. Syst.* 42(3-4):177–190.

Fitzgerald, T.; Goel, A.; and Thomaz, A. 2017. Human-robot co-creativity: Task transfer on a spectrum of similarity. In Goel, A.; Jordanous, A.; and Pease, A., eds., *Proc. 8th Int. Conf. Computational Creativity*, 104–111.

Froese, T., and Ziemke, T. 2009. Enactive artificial intelligence: Investigating the systemic organization of life and mind. *Artificial Intelligence* 173(3):466–500.

Gemeinboeck, P., and Saunders, R. 2014. Towards a performative body mapping approach. In *Proc. 50th Annual Convention of the AISB*. London: Society for the Study of Artificial Intelligence and the Simulation of Behaviour.

Gemeinboeck, P. 2017. Introduction to FCJ-203 Creative Robotics: Rethinking human machine configurations. *Fibre Culture* 203:1–7.

Glăveanu, V. P. 2017. Creativity in craft. In Kaufman, J. C.; Glăveanu, V. P.; and Baer, J., eds., *The Cambridge Handbook of Creativity Across Domains*. Cambridge, UK: Cambridge University Press.

Graves, A. 2014. Generating sequences with recurrent neural networks. Technical Report arXiv:1308.0850v5 [cs.NE].

Guckelsberger, C.; Salge, C.; and Colton, S. 2017. Addressing the "why?" in computational creativity: A non-anthropocentric, minimal model of intentional creative agency. In *Proc. 8th Int. Conf. Computational Creativity*.

Heider, F., and Simmel, M. 1944. An experimental study of apparent behavior. *The American Journal of Psychology* 57(2):243–259.

Koh, J. T.; Dunstan, B. J.; Silvera-Tawil, D.; and Velonaki, M., eds. 2016. *Cultural Robotics: First International Workshop, CR 2015, IEEE RO-MAN 2015, Kobe, Japan, August 31, 2015. Revised Selected Papers*, volume 9549 of *LNAI*. Springer.

Lakoff, G., and Johnson, M. 1999. *Philosophy in the Flesh: The Embodied Mind and Its Challenge to Western Thought*. Collection of Jamie and Michael Kassler. Basic Books.

Levillain, F., and Zibetti, E. 2017. Behavioral objects: The rise of the evocative machines. *Journal of Human-Robot Interaction* 6(1):4–24.

Malafouris, L. 2014. Creative thinging: The feeling of and for clay. *Pragmatics & Cognition* 22(1):140–158.

Michotte, A. 1963. *The Perception of Causality*. Oxford, England: Basic Books.

Noë, A. 2004. *Action in Perception*. Cambridge, MA: MIT Press.

Nunnally, J. 1978. *Psychometric theory*. New York: McGraw-Hill, 2nd edition.

Özen, F.; Tükel, D. B.; and Dimirovski, G. 2017. Synchronized dancing of an industrial manipulator and humans with arbitrary music. *Acta Polytechnica Hungarica* 14(2):151–169.

Penny, S. 2000. Agents as artworks and agent design as artistic practice. In Dautenhahn, K., ed., *Human Cognition and Social Agent Technology*. Amsterdam, NL: John Benjamins. 395–413.

Pin, F. G., and Killough, S. M. 1994. A new family of omnidirectional and holonomic wheeled platforms for mobile robots. *IEEE transactions on robotics and automation* 10(4):480–489.

Reynolds, D. 2012. Kinesthetic engagement: Embodied responses and intersubjectivity: Introduction. In Reynolds, D., and Reason, M., eds., *Kinesthetic empathy in creative and cultural practices*. London: Intellect Books. 87–90.

Ros, R.; Baroni, I.; and Demiris, Y. 2014. Adaptive human–robot interaction in sensorimotor task instruction: From human to robot dance tutors. *Robot. Auton. Syst.* 62:707–720.

Saegusa, R.; Metta, G.; Sandini, G.; and Sakka, S. 2009. Active motor babbling for sensorimotor learning. In *Proc. IEEE Int. Conf. Robotics and Biomimetics, 2008*, 794–799.

Sawyer, R. K. 1998. The interdisciplinary study of creativity in performance. *Creativity Research Journal* 11(1):11–19.

Scholl, B. J., and Tremoulet, P. D. 2000. Perceptual causality and animacy. *Trends in Cognitive Sciences* 4(8):299 – 309.

Schön, D. A. 1983. *The reflective practitioner : how professionals think in action*. New York, NY: Basic Books.

Stevens, C., and McKechnie, S. 2005. Thinking in action: thought made visible in contemporary dance. *Cognitive Processing* 6:243–252.

Stewart, D. 1965. A platform with six degrees of freedom. *Proc. Inst. Mechanical Engineers* 180(1):371–386.

Suschke, S. 2003. *Müller macht Theater: Zehn Inszenierungen und ein Epilog*. Berlin: Theater der Zeit.

Takala, T. 2015. Preconceptual creativity. In Toivonen, H.; Colton, S.; Cook, M.; and Ventura, D., eds., *Proc. 6th Int. Conf. Computational Creativity*, 252–259.

Tukey, J. W. 1977. *Exploratory Data Analysis*. Addison-Wesley.

Varela, F. J.; Thompson, E.; and Rosch, E. 1991. *The Embodied Mind: Cognitive Science and Human Experience*. Cambridge, MA: MIT Press.

Vlachos, E., and Scharfe, H. 2015. An open-ended approach to evaluating android faces. In *The 24th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN 2015)*, 746–751.

Walters, M.; Marcos, S.; Syrdal, D.; and Dautenhahn, K. 2013. An interactive game with a robot: People's perceptions of robot faces and a gesture based user interface. In *Proc. 6th Int. Conf. Advanced Computer-Human Interactions*, 123–128.

Ziemke, T. 2016. The body of knowledge: On the role of the living body in grounding embodied cognition. *BioSystems* 148:4–11.

Conceptualising Computational Creativity: Towards automated historiography of the research field

Vid Podpečan[1], **Nada Lavrač**[1], **Geraint Wiggins**[2,3], **Senja Pollak**[1,4]
[1] Department of Knowledge Technology, Jožef Stefan Institute, Ljubljana, Slovenia
[2] Computational Creativity Laboratory, AI Lab, Vrije Universiteit Brussel, Belgium
[3] School of Electronic Engineering and Computer Science, Queen Mary University of London, UK
[4] USHER Institute, University of Edinburgh, Edinburgh, UK
{vid.podpecan,nada.lavrac,senja.pollak}@ijs.si; geraint.wiggins@vub.be

Abstract

This paper reports on the progress towards constructing automated historiography of the research field of Computational Creativity (CC). The analysis is based on papers published in the Proceedings of International Conferences on Computational Creativity in eight consecutive years since 2010. This paper extends our earlier work by proposing an approach to CC field analysis that facilitates the automation of CC conceptualisation.

Introduction

Computational Creativity (CC) is concerned with engineering software that exhibits behaviours that would reasonably be deemed creative (Boden, 2004; Colton and Wiggins, 2012). As for every other research community, it is crucial for the CC community to analyse its research topics, applications and the overall progress of the field with the goal of CC field conceptualisation.[1]

Loughran and O'Neill (2017) have studied the CC domain by analysing its conferences and proceedings, where—as they acknowledge—conceptual categorisation was conducted subjectively, through a review of each paper. In contrast, the aim of the research presented in the current paper is to provide an semi-automated analysis of the field as it develops, with the expectation that this may be used in the future for automated construction of the historiography of CC research, which can substitute or complement manual analysis of the research field. Our long term vision is to provide a system, which would be fully automated and available online to the CC community for its analysis and promotion to a wider public.

The conceptualization of the CC research field has been studied already in our past research, where a mixture of text analysis and clustering methods was used (Pollak et al., 2016). In this paper we report on further work in this direction, complementing the previous study by introducing an extended set of methods and by analysing papers published in additional ICCC proceedings. We show how the extended set of methods can be used to support the understanding of the conceptual structure of the field as represented by the papers presented at its annual International Conference on Computational Creativity (ICCC).

The paper is structured as follows. First, we briefly review the previous attempt to address this question. Next, we describe the data used in the study, followed by the section in which we explain the methodology that (a) supports topic analysis through diachronic clustering, (b) uses a contemporary visualisation method, and (c) involves relatively little human intervention, to the extent that can be fully automated in the future. We present the results of this methodology and explain the achieved conceptualisation.

Experimental data

We used the ICCC corpus presented by Pollak et al. (2016) constituting of the articles from the proceedings of the 2010–2015 International Conferences on Computational Creativity, and complemented it with the papers from the years 2016–2017. The text files were converted from PDF to TXT and the bibliography sections were removed. Our corpus consists of 340 articles in total (see Figure 1).[2]

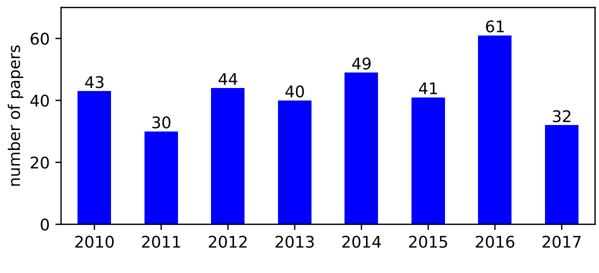


Figure 1: Numbers of papers in the ICCC Proceedings.

[1] We use the term conceptualisation in alignment with its standard use in information science, where conceptualisation is defined as an abstract (simplified) view of some selected part of the world, containing the objects, concepts, and other entities that are presumed of interest for some particular purpose and the relationships between them (Gruber, 1993; Smith, 2003).

[2] Note that there might be minor differences between the number of articles in the corpus and the actual proceedings, since for 2010–2015 the corpus was manually collected and we cannot exclude human mistakes, while for 2016 and 2017 it has been crawled automatically, but we have noticed a few document duplicates.

Previous results

In our previous work we performed domain conceptualisation by applying semi-automated, user-guided clustering using a topic ontology construction tool OntoGen (Fortuna, Mladenič, and Grobelnik, 2006). The resulting corpus-based categorisation of the field of Computational Creativity, presented in detail by Pollak et al. (2016), identified the following main subdomains of Computational Creativity: Musical, Visual, Linguistic creativity, Games and creativity, Conceptual creativity, as well as the domain of Evaluation, which was added after a manual query used in the active learning approach to topic ontology creation. For several subdomains, subcategories were detected at a lower level, including Narratives, Poetry, Recipes and Lexical creativity as subdomains of Linguistic creativity.

Proposed Domain Conceptualisation Methodology

The main ingredients of the extended methodology are: detection of CC topics by document clustering, enrichment of the analysis by cluster visualisation, and performing clustering incrementally on different datasets, starting with the first edition of ICCC 2010, and finally using the entire ICCC proceedings data set (2010–2017) in the final analysis, thus mimicking the continuous automatic analysis support that we aim to make available to the community in the future.

Data cleaning and preprocessing

First, we have performed a number of preprocessing steps in order to make the data suitable for the analysis. One by one, the articles from the ICCC corpus were sent to the following pipeline to obtain lists of tokens:

1. decode all characters from UTF-8 to ASCII using the Unidecode library[3] in order to remove some of the artifacts introduced by the PDF-to-text conversion;
2. split sentences using the Punkt tokeniser (Kiss and Strunk, 2006);
3. expand contractions;
4. word tokenisation using the Treebank tokeniser (Marcus et al., 1994);
5. token filtering to remove tokens of length less than two, unprintable tokens, numbers, and non-alphanumeric tokens;
6. lemmatisation using the LemmaGen lemmatiser (Juršič et al., 2010);
7. adding bi-grams and tri-grams;
8. removing stopwords.

In spite of elaborate automated preprocesing to remove PDF-to-text artefacts, several smaller issues such as hyphenation, ligatures etc. remain and can be observed in some of the visualizations. For example, the character sequence *tion* is a common ending of several hyphenated words and thus appears as an important term in several wordclouds.

[3]Unidecode is based on hand-tuned character mappings that also contain ASCII approximations for symbols and non-Latin alphabets.

Diachronic paper grouping

This research aims to provide a methodology for continuous, automated historiography of the field. In this setting, after each conference, the editors would upload the papers to the system, and the clustering (topic identification) would be automatically produced. The resulting information, essentially a set of topological representations, can then be used computationally to create descriptions of all or part of the field.

For this reason, we group the papers cumulatively by year, starting with the first edition of ICCC, year y1=2010, then adding the next year's proceedings to the corpus in year 2 (y2=2010–2011), and so on. The latest set of documents consists of all the available papers (y8=2010–2017).

Clustering

In order to perform document clustering, vectors of tokens as returned by the preprocessing pipeline described above were first transformed into tf·idf vectors (Term Frequency·Inverse Document Frequency: Salton and Buckley, 1988). This was followed by Latent Semantic Indexing (LSI) (Deerwester, 1988), which performs singular value decomposition and keeps only the largest values thus effectively reducing the dimensionality by several orders of magnitude and reducing noise.

In general, determining the optimal number of target dimensions when performing LSI is still a challenge. For a real world sized corpus with e.g., 10^5 documents, a number such as 300 is considered as appropriate (Bradford, 2008). Taking into account that our corpus consists of only 340 articles, we have set the desired number of dimensions to 10 after a series of experiments where the *silhouette score* (Rousseeuw, 1987) was measured for a different number of target dimensions and a different number of clusters. When the number of LSI dimensions was around 10, the silhouette score did not show anomalous trends such as monotone increasing or decreasing and visualization of the corpus revealed clearly visible groups of data points, which

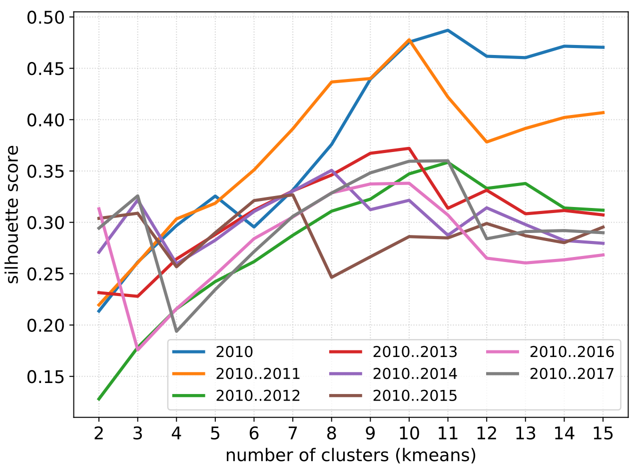


Figure 2: Silhouette scores for different values of *k* and different (cumulative) document sets.

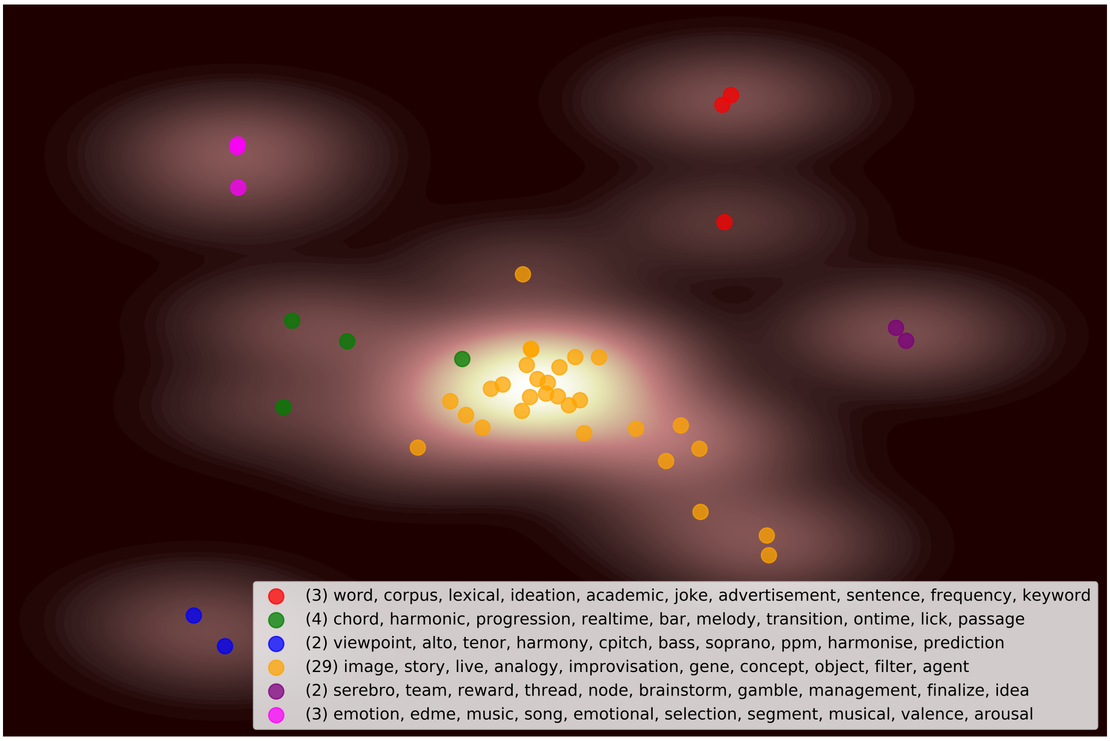


Figure 3: A visualisation of the ICCC 2010 proceedings papers clustered into 6 clusters.

indicates that such setting is appropriate for revealing the structure of the corpus and for reducing the noise introduced by text extraction.

Since we are interested in fully automated methods, we experimented first with the DBSCAN clustering algorithm (Ester et al., 1996) that does not require the number of clusters set as a parameter. However, the results were poor and the algorithm was not able to find dense regions in the data. Therefore, we resorted to the *k-means* clustering algorithm. For manually selecting the k parameter (the number of clusters), the user can rely on the visualisation of the document space and expert knowledge of the domain. In addition, we have evaluated the silhouette score for 2 to 15 clusters (see Figure 2) in order to investigate whether the optimal number of clusters can be determined automatically. We compared the results of manual and silhouette-based k setting, and decided to focus on the results with manually set k as they led to more meaningful interpretations. We defer automatic parameter setting experimentation to future work.

Finally, the results of clustering were also used to automatically extract keywords. For each cluster, the top t terms (tokens) of the mean tf·idf vector (centroid) are collected and presented to the user. The terms can be used to identify main topics, the diversity of the cluster, detect outliers and evaluate whether the number of clusters is appropriate.

Visualisation

We have devised a visualisation methodology to support historiographic analysis of a given domain described by a set of documents and a timeline. The methodology consists of (a) combining the results of clustering with a 2D projection of the document space, (b) wordclouds and (c) composition of the result of (a) from different time points into a video clip.

2D visualisation First, the documents are preprocessed and LSI vectors are computed and clustered as described in the preceding subsection. Then, an Isomap (Tenenbaum, Silva, and Langford, 2000) projection is computed which yields 2D coordinates for each document. Using Isomap results we draw a scatterplot where each point represents one document. In addition, the cluster index is used to assign colours to points, while top-weighted centroid terms (keywords) are used for cluster summarisation[4]. On top of that, we use a 2D kernel density estimation to compute the shading of the scatterplot background. This visualisation is shown in Figure 3.

Wordclouds In addition to the 2D map of the corpus we also compute and display wordclouds that can help in identifying the keywords and topics of the selected document set (which can be either the current year or a cumulative set of all years up to the current time point). Figures 4 and 5 show wordclouds for the first and the last year of the ICCC

[4]In all the presented figures only 10 keywords per cluster are shown due to limited figure width.

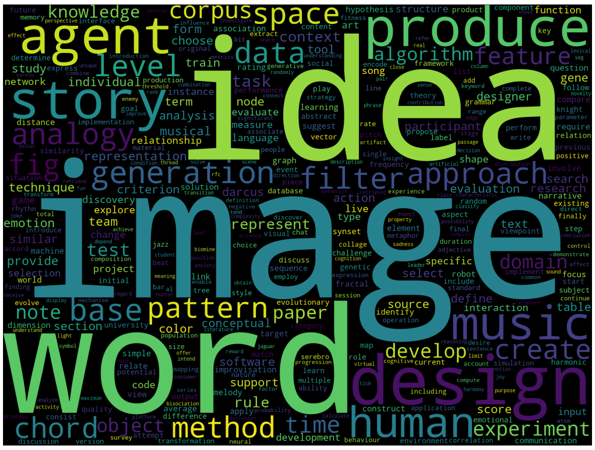


Figure 4: A filtered wordcloud of the 2010 ICCC articles.

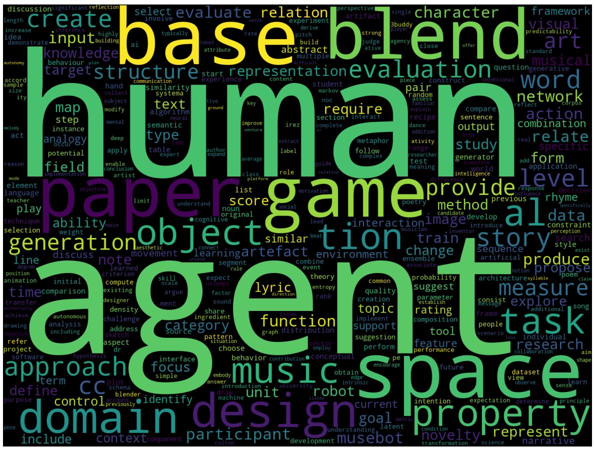


Figure 5: A filtered wordcloud of the 2017 ICCC articles.

proceedings corpus with the following most frequent general terms manually removed prior to wordcloud drawing: *creativity, creative, model, process, computational, result, generate, concept, set, figure*.

Animation The described 2D visualisation procedure can be used to create animations of the changes of the document space through time. Such animation can be used to follow the development of topics through time, observe merging and splitting and detect trends. To generate a movie, the visualisation procedure is applied sequentially to a growing collection of documents. In each time step, new documents are added and a new image is produced. Finally, the pictures are merged in a video clip and the crossfade effect is applied to smooth the transitions from one image to another.

We have also implemented a modification that enables tracking of topics/clusters. By default, colours for clusters (data points) are selected randomly. This is sufficient for single images but may introduce confusion when several images are merged into a video because a cluster of a certain colour is not necessarily related to a cluster of the same colour on the next video frame. Therefore, we have implemented a heuristic approach that works as follows. For each time step we compare the current clustering keywords with the previous ones. Whenever a high level of similarity between two ranked keyword lists is detected we assume that this is the same cluster so the same colour will be used in the current image. In addition, we change the shape of the scatterplot points to allow for visual detection of such clusters. The similarity between two ranked lists is computed using the Rank-biased overlap algorithm (Webber, Moffat, and Zobel, 2010) and was found to reliably detect similarities between ordered lists.

Results: ICCC Topics Across the Years

We analysed the results for different values of k and different sets of years. For example, if we input the papers of the first edition of ICCC in 2010, and make a single split into 2 clusters (Figure 6), we see that the documents are grouped into *Musical creativity*, with the keywords *music, melody, harmonic, song*, while the other cluster comprises all other themes. For deeper understanding, we can see the title of the documents, and the paper with the title "User-Controlling Expressed Emotions in Music with EDME" explains the keywords *edme* and *emotion* in the cluster name. The clustering probably illustrates the familiar problem of disjoint terminology between musical creativity papers and others.

If we set k higher, we can get a more realistic topic overview. For instance in Figure 3 (same document set, split into 6 clusters), *Musical creativity* can be observed across several clusters, one related to the modeling of harmony (blue), while others cover the papers related to emotions and music (pink in upper left corner with documents "Real-Time Emotion-Driven Music Engine", "Automatic Generation of Music for Inducing Emotive Response" and "User-Controlling Expressed Emotions in Music with EDME") and the green to the generation of harmonic progression and jazz.

The clustering with the highest silhouette score for 2010 was $k = 11$ (see silhouette score comparison in Figure 2). In Figure 7, we can see that since the corpus is small, the clusters contain very few documents, but the several topics (that will appear also in the expanded datasets with the consecutive years) are announced, such as *lexical creativity-story generation* (keywords: story, knight, narrative, jaguar), *reasoning/association/bisociation* (papers: "Constructing Conceptual Spaces for Novel Associations", "Bisociative Knowledge Discovery"…, "Domain Bridging Associations" and "Some Aspects of Analogical Reasoning in Mathematical Creativity"). The largest cluster refers to *visual creativity*, with the keywords: image, filter, darcus, robot, collage, fractal, but also several keywords due to noisy clustering—e.g., chat. (The term *darcus* is a lemmatised version of the system DARCI (mistakenly but reasonably assigning to the term a Latin origin)). The papers comprise "The Painting Fool", "Swarm Painting Atelier", "A Fractal Approach Towards Visual Analogy".

Over the years, the clustering becomes more interesting, since we have more documents. So, for instance, in the vi-

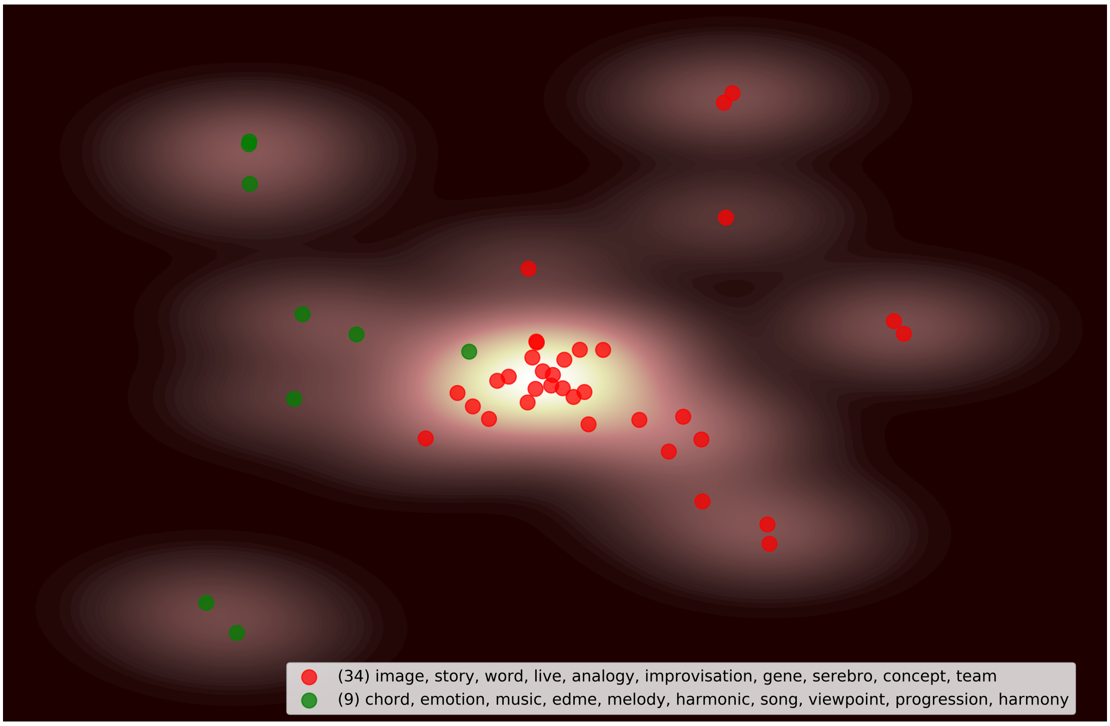


Figure 6: ICCC 2010 proceedings papers clustered into 2 clusters.

sualisation of papers from 2010–2015 in Figure 8, the CC domains are very clearly separated and characterised by corresponding keywords. *Music* is visible in yellow, *stories* and *games* are corresponding to the blue and green, respectively, in red we have *poetry* (keywords: poem, poetry, flowchart, syllable, rhyme, word, bengali, flowr), while in purple are other documents, including a clear coverage of *image*.

Unsurprisingly, the most interesting is the topic clustering on the entire corpus. The highest silhouette scores were returned for k=10 and k=11. We first analyse the k=10 clustering results. Since it covers the ICCC conceptualisation with the entire paper selection, we describe it in more detail, in terms of our category name and the associated keywords:

1. **Poetry** poem, poetry, flowchart, rhyme, syllable, word, bengali, expert, tweet, grammar, simile, template, constraint, text, poetryme
2. **Games** game, angelina, player, mechanic, utterance, jam, miner, mechanics, gameplay, rogue, spaceship, play, suspect, agent, designer
3. **Concepts** blend, icon, i1, blender, amalgam, conceptual, ontology, i2, colimit, space, optimality, goguen, workflow, input, relation
4. **Music** musical, music, chord, improvisation, musebot, melody, accompaniment, pitch, jazz, lyric, composition, musician, harmonic, song, participant
5. **Story and Narrative** story, character, narrative, knight, jaguar, action, plot, mexica, tension, enemy, princess, event, storyteller, scene, rez
6. **Image** darcus, image, adjective, synset, rendering, painting, fool, artifact, pareidolia, icon, volunteer, fiery, association, peaceful, train
7. **Embodiment and Choreography** robot, dancer, movement, choreographer, dance, empowerment, choreography, motion, agent, robotic, embody, antagonistic, sensor, choreographic, keyframe
8. **Cuisine** artifact, recipe, ingredient, surprise, novelty, rdc, haiku, card, apparel, artefact, cocktail, expectation, maze, inspiring, regression
9. **Conceptualising CC** cc, id, mlcc, copula, additive, artifact, preference, attribute, iccc, gaver, ig, function, student, marginal, intentional
10. **Other (not classified)** image, agent, object, node, association, analogy, word, metaphor, shape, painting, concept, conceptual, software, fitness, fig

As can be seen from the keywords, some clusters are pure while others include noise. We have yet to perform an extensive cluster evaluation, but we provide in Table 1 a full list of documents for the first topic cluster, where the precision is very high. The papers are available on the web[5], so minimal references are given here.

In addition, we compared the results of k=10 to k=11, to see if the clustering results are stable. We have observed that the *Poetry* cluster remains exactly the same (contains the same papers). The same holds true for the cluster (*Games*),

[5] `http://computationalcreativity.net`

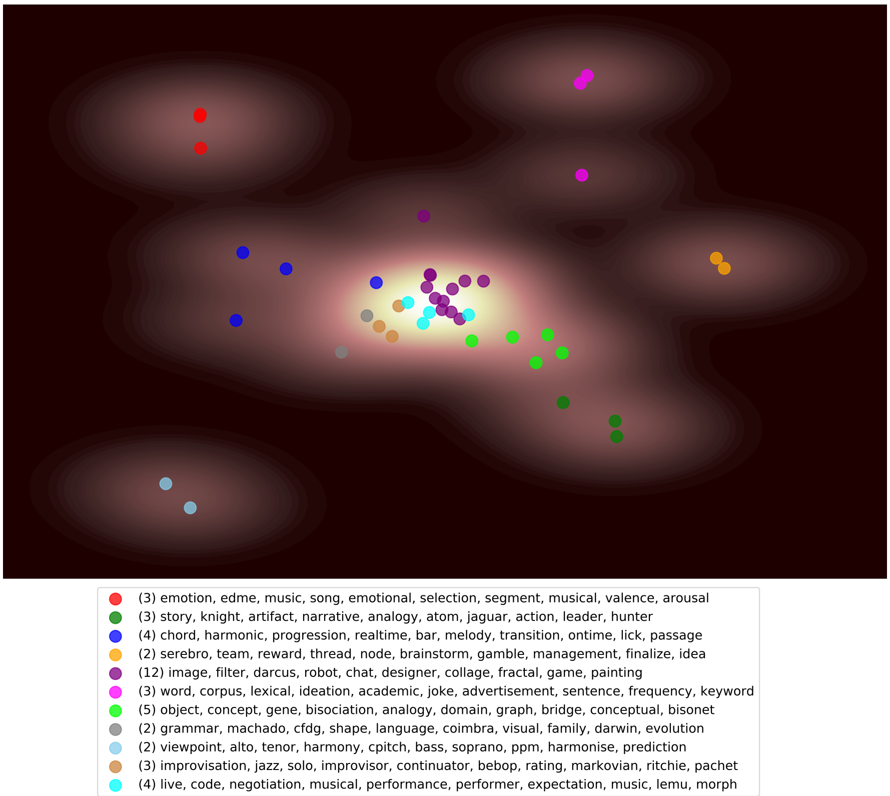


Figure 7: ICCC 2010 proceedings papers clustered into 11 clusters.

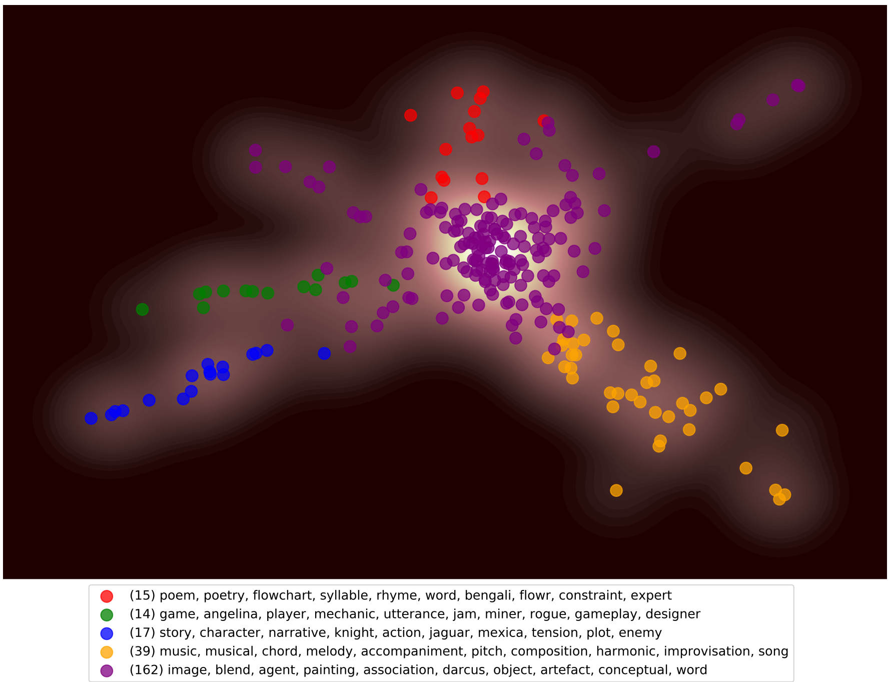


Figure 8: Papers of 2010–2015 ICCC proceedings clustered into 5 clusters.

the cluster related to conceptual blending (cluster *Concepts*), *Story and Narratives* and *Embodiment and Choreography*. Also the *Cuisine* cluster remains the same, but covers some papers, which do not belong to this cluster. For instance, the paper by Dan Ventura "Mere Generation: Essential Barometer or Dated Concept?" questions general principles of creativity, where recipes are just one of the several examples that are used in discussion. In Visual creativity (cluster Images), the only difference is in an additional paper added to the cluster with k=11, which is the paper describing the event You Cant Know my Mind. The cluster *Conceptualising CC*, contains one more paper in the cluster of k=10, which is in k=11 unclassified: this is our CC conceptualisation attempt (Pollak et al., 2016), for which it is understandable that it is not fixed to a single cluster, since it discusses different topics of computational creativity. The biggest difference can be observed in the *Musical* cluster, which is in the setting with k=11 split into two distinct clusters, with the following keywords:

- Music-C1: *musebot, musical, agency, improvisation, musician, music, jazz, participant, interaction, ensemble, performer, bown, improvise, kelly, practice*
- Music-C2: *music, chord, musical, melody, accompaniment, lyric, pitch, harmonic, composition, song, audio, markov, edme, beat, bass*

The conceptualisation across the years provides the clustering where the papers in each cluster can be used as reading material for the new members joining the ICCC community and being especially interested in a specific subdomain.

Conclusions and Future Work

This paper presents an overview of ICCC proceedings topics, achieved by the proposed methodology composed of data preprocessing, clustering and cluster visualisation. Since computational creativity is still a relatively new research field, it is still possible for the researcher to have an overview of the field as a whole, but with the growth of the field this will no longer be possible. Therefore, it is useful to provide a transparent and accessible overview of topics and categorised papers for sub-domains. We consider that this is very important especially for the incomers to the field.

We presented the results of analyzing different document sets and found out that the clustering results are mostly meaningful, allowing the expert to easily recognise the topics (e.g., musical creativity, story generation, poetry generation, visual creativity, culinary creativity, conceptual creativity, etc.). We experimented with automated discovery of the optimal number of cluster using the silhouette score but so far the results were not conclusive, since they did not fully align with human observations using 2D visual representations.

We will continue to work towards the automation of the process including clustering, concept naming, tracking topic changes within the selected domains, and computationally creating correct narratives over the history of computational creativity.

Table 1: ICCC papers captured in Cluster 1: Poetry

Bay, B., Bodily, P., and Ventura, D. (2017). Text transformation via constraints and word embedding.

Charnley, J., Colton, S., and Llano, M. T. (2014). The FloWr Framework: Automated Flowchart Construction, Optimisation, Alteration for Creative Systems.

Colton, S. and Charnley, J. (2013). Towards a flowcharting system for automated process invention.

Colton, S., Goodwin, J., and Veale, T. (2012). Full-FACE poetry generation.

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.

Das, A. and Gambäck, B. (2014). Poetic Machine: Computational Creativity for Automatic Poetry Generation in Bengali.

Gervás, P. (2011). Dynamic inspiring sets for sustained novelty in poetry generation.

Gross, O., Toivanen, J. M., Lääne, S., and Toivonen, H. (2014). Arts, News, Poetry — The Art of Framing.

Kantosalo, A., Toivanen, J. M., and Toivonen, H. (2015). Interaction evaluation for human-computer co-creativity: A case study.

Lamb, C., Brown, D. G., and Clarke, C. (2015). Human competence in creativity evaluation.

Lamb, C., Brown, D. G., and Clarke, C. L. (2017). Incorporating novelty, meaning, reaction and craft into computational poetry: a negative experimental result.

Lamb, C., Brown, D. G., and Clarke, C. L. A. (2016). Evaluating digital poetry: Insights from the cat.

Misztal, J. and Indurkhya, B. (2014). Poetry generation system with an emotional personality.

Oliveira, H. G., Hervás, R., D'ıaz, A., and Gervás, P. (2014). Adapting a Generic Platform for Poetry Generation to Produce Spanish Poems.

Oliveira1, H. G. and Alves, A. O. (2016). Poetry from concept maps – yet another adaptation of poetryme's flexible architecture.

Rashel, F. and Manurung, R. (2014). Pemuisi: a constraint satisfaction-based generator of topical Indonesian poetry.

Tobing, B. C. and Manurung, R. (2015). A chart generation system for topical metrical poetry.

Toivanen, J. M., Järvisalo, M., and Toivonen, H. (2013). Harnessing constraint programming for poetry composition.

Toivanen, J. M., Toivonen, H., Valitutti, A., and Gross, O. (2012). Corpus-Based generation of content and form in poetry.

Acknowledgements

The work of VP, NL and SP was partly funded by the research programme Knowledge Technologies, funded by the Slovenian Research Agency. GW very gratefully acknowledges the financial and academic support of the Jožef Stefan Institute, during his sabbatical there in October and November 2017.

References

Boden, M. A. 2004. *The Creative Mind: Myths and Mechanisms*. Routledge.

Bradford, R. B. 2008. An empirical study of required dimensionality for large-scale latent semantic indexing applications. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management*, CIKM '08, 153–162. New York, NY, USA: ACM.

Colton, S., and Wiggins, G. A. 2012. Computational creativity: The final frontier? In *Proceedings of the 20th European Conference on Artificial Intelligence*, ECAI'12, 21–26. Amsterdam, The Netherlands, The Netherlands: IOS Press.

Deerwester, S. 1988. Improving Information Retrieval with Latent Semantic Indexing. In Borgman, C. L., and Pai, E. Y. H., eds., *Proceedings of the 51st ASIS Annual Meeting (ASIS '88)*, volume 25. Atlanta, Georgia: American Society for Information Science.

Ester, M.; Kriegel, H.-P.; Sander, J.; and Xu, X. 1996. A density-based algorithm for discovering clusters a density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, KDD'96, 226–231. AAAI Press.

Fortuna, B.; Mladenič, D.; and Grobelnik, M. 2006. Semi-automatic construction of topic ontologies. In *Semantics, Web and Mining: Joint International Workshops, EWMF 2005 and KDO 2005, Revised Selected Papers*, 121–131. Springer.

Gruber, T. R. 1993. A translation approach to portable ontology specifications. *Knowledge Acquisition* 5(2):199–220.

Juršič, M.; Mozetič, I.; Erjavec, T.; and Lavrač, N. 2010. Lemmagen: Multilingual lemmatisation with induced ripple-down rules. *J. univers. comput. sci.* 16:1190–1214.

Kiss, T., and Strunk, J. 2006. Unsupervised multilingual sentence boundary detection. *Computational Linguistics* 32(4):485–525.

Loughran, R., and O'Neill, M. 2017. Application domains considered in computational creativity. In *Proceedings of ICCC 2017*. Association for Computational Creativity.

Marcus, M.; Kim, G.; Marcinkiewicz, M. A.; MacIntyre, R.; Bies, A.; Ferguson, M.; Katz, K.; and Schasberger, B. 1994. The penn treebank: Annotating predicate argument structure. In *Proceedings of the Workshop on Human Language Technology*, HLT '94, 114–119. Stroudsburg, PA, USA: Association for Computational Linguistics.

Pollak, S.; Boshkoska, B. M.; Miljkovic, D.; Wiggins, G.; and Lavrač, N. 2016. Computational creativity conceptualisation grounded on iccc papers. In François Pachet, Amilcar Cardoso, V. C. F. a. G., ed., *Proceedings of ICCC 2016*, 123–130. Association for Computaitonal Creativity.

Rousseeuw, P. J. 1987. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics* 20:53 – 65.

Salton, G., and Buckley, C. 1988. Term-weighting approaches in automatic text retrieval. *Information Processing & Management* 24(5):513–523.

Smith, B. 2003. Chapter 11: Ontology. In Floridi, L., ed., *Blackwell Guide to the Philosophy of Computing and Information*, volume 7250. Blackwell. 155–166.

Tenenbaum, J. B.; Silva, V. d.; and Langford, J. C. 2000. A global geometric framework for nonlinear dimensionality reduction. *Science* 290(5500):2319–2323.

Webber, W.; Moffat, A.; and Zobel, J. 2010. A similarity measure for indefinite rankings. *ACM Trans. Inf. Syst.* 28(4):20:1–20:38.

The Image Artist: Computer Generated Art Based on Musical Input

Paper type: System Description Paper

Viktor Zorić and Björn Gambäck
Department of Computer Science
Norwegian University of Science and Technology
7491 Trondheim, Norway
viktor.zoric@outlook.com gamback@ntnu.no

Abstract

The paper explores computer-generated art based on musical input, using evolutionary algorithms (EA) for a music-to-image transformation which operates on music and image metadata, such as music and image features, rather than raw data. The metadata is utilized through a novel usage of mapping tables that work as recipes for images. The mapping tables are evolved without interactive involvement. Experiments were carried out on using the mapping tables to match music and image features, and with various fitness functions that combined user preferences with art novelty aspects. Fitness functions based on a distance measure between user preferred mapping tables and the computer-generated mapping tables can efficiently drive the EA towards the user's preference. Novel and interesting mappings between the image and music features can be achieved with fitness functions that selectively ignore some user preferences. Evolution of palettes and figure parameters can match user expectation without knowing the structure of the optimal mapping table.

1 Introduction

Visual art can enrich various aspects, phases and situations in human life, with the support role for computing art being situation dependent. The topic of this work is to create a system that can generate image art based on musical input with use of evolutionary algorithms (EA). This means that the system should create images that share features with the corresponding music. Such features could be emotional or artistic, where the aim is to create a correlation between image and music that the end user agrees on. The system should be able to take an arbitrary musical piece as input and create a corresponding image, considering both end user specifications and novelty.

The work combines a theoretical and a practical approach. It is a design, implementation and experiments-driven exercise, where end-user involvement — survey results and user interaction tests — contributed to the EA functionality. The EA for music-to-image transformation operates on music and image metadata (attributes), rather than raw data. The metadata (for music and image features) is utilized by mapping tables that work as recipes for images. The algorithm generates images by evolving the mapping tables without interactive involvement. Using metadata and mapping tables in evolutionary algorithms introduces an alternative approach to computer generated image art, compared to previous research.

The next section introduces some work that has inspired the present project. Then Section 3 describes the system architecture, while Section 4 shows some experiments using the system. In Section 5 a discussion of the system is presented, and possible future work outlined.

2 Related Work

Over the last two decades there have been many different approaches to generation of art using computers, with evolutionary algorithms being a recurring method. EAs are highly dependent on a fitness function which accurately describes, in mathematical terms, how good a solution is. Lacking this feature the algorithm will struggle to generate a good solution set. But evaluating aesthetics and art is a subjective process, so a well performing general mathematical fitness function for art is absent. Instead of writing functions that find subjectively good looking patterns in image art, several approaches to generative art programs thus use interactive search methods, where humans take part in the evaluation of aesthetics/quality. In interactive evolutionary algorithms (Sims, 1991), humans must evaluate solutions through subjective judgement (Eiben & Smith, 2015), that the algorithm can use to generate offspring, by setting the fitness value of each solution, or by selecting phenotypes to mate. Following Todd & Latham (1992) interactive evolutionary computing dominated the evolutionary art field in the 1990's, with the vast majority of the 200 citations cataloged by Lewis (2008) using some form of case-by-case human judgment.

Ashmore & Miller (2004) stress that the main difference between evolutionary art and other search problems is that the fitness of an image is based on something that is very hard to describe or maybe even to understand, since the attractiveness of an image is personal and differs among people. With evolutionary art, the search is more exploratory, with divergence and diversity being the key factors. Understanding the nature of visual representation requires asking what artists need to know in order to make representational objects; knowledge not only about the world, but also about the nature and the strategies of representation (Cohen, 1988). Woolley & Stanley (2011) showed that the used representation has a major impact on the evolution of images (including performance). Given the hardness of this kind of application, it would be desirable to have representations that have high locality (Galván-López et al., 2011), so that small changes to the genotype correspond to small changes to phenotype. Johnson (2016) classified a large collection of research using a taxonomy of the ways in which fitness is used in EA art and music systems, with two dimensions: what the fitness function is applied to and the basis by which the function is constructed.

Significant here are the analyses of Machado & Cardoso (2002), Baluja et al. (1994) and Kowaliw et al. (2009) that present various techniques to overcome the limitations of interactive EAs. Secretan et al. (2011) and Clune & Lipson (2011) use web-based interactive EA to let users evolve lineages of artifacts based on their preferences, rate evolved artifacts, and take previously evolved artifacts and continue the search process themselves, so that artifacts can be the product of a collaborative search process. The present work will try to make a compromise by using the results of both user interaction and automated computing based on fine-tuned fitness functions to steer evolutionary algorithms. This complementarity can possibly offer both promising artistic results and convergent algorithms.

One of the key aspects of the evolutionary art is the novelty. Lehman & Stanley (2010) proposed a novelty search algorithm for evaluating image uniqueness. For each image, a novelty score is computed, taking into account its neighbours and an archive containing the most novel images. Vinhas et al. (2016) explore the effects of introducing novelty search in evolutionary art (they define novelty as phenotypic diversity). Their algorithm combines fitness and novelty metrics to frame image evolution as a multi-objective optimization problem, promoting the creation of images that are both suitable and diverse. Dumoulin et al. (2016) investigate the construction of a single, scalable deep network that can capture the artistic style of a diversity of paintings. They claim that their model permits a user to explore new painting styles by arbitrarily combining the styles learned from individual paintings.

Two projects have particularly inspired the present work: *The Painting Fool* (Colton, 2012) creates art by simulating natural painting strokes of different types through parameterization, which allows for the discovery of novel painting styles, combined with a database of mappings between emotions and different styles, to alter some styles by enhancing given emotions. Evolutionary algorithms are also implemented to expand the abilities to create novel results. *Sågawave* (Bredesen et al., 2016) focuses on creating images from songs, using Spotify API to fetch songs, Web Audio API to analyze them, and React front end library to draw images. Images are generated while the music is playing, and drawn from left to right as the songs progress. Frequency values determine how many shapes there will be and where on the canvas they are drawn. Amplitude values are used to select shape colours, while number of beats per minute map to weighting of colours and whether to draw sharp edged objects or not.

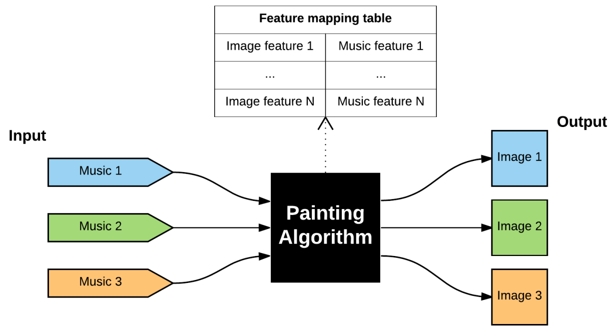


Figure 1: The Workflow of the Painting Algorithm

3 Architecture

This section discusses the architecture and design of the system for Music-to-Image transformation using Evolutionary Algorithms. The EAs are implemented from scratch to have full control of the evolution and its strategies. The whole framework is written in Java, utilizing Java's built in graphical library, and with a custom written interface for fetching music features using Spotify API. Spotify is a music streaming service that provides access to millions of songs that have downloadable previews containing audio analysis data.

Evolutionary Algorithms depend on three main parts: the genotype, the phenotype, and the evolutionary loop. The evolutionary loop is further guided by a fitness function. Creating images from music requires a mapping between music and image features. Music features are obtained through Spotify API, while image features need to be created during the act of painting using available painting tools. Figure 1 illustrates the workflow of the painting algorithm. The **phenotype** in this work will be called "The Image Artist" and produces a set of images, based on the **genotype** (represented by the feature mapping table). For each input song, a corresponding image is generated by using this feature mapping table. It is a metadata set (with key-value pairs) used to create painting instructions.

The evolutionary algorithms use the music indirectly, i.e., via various music parameters / descriptors, also denoted as metadata. Rather than running audio analysis on raw data, which requires the client to possess the music file, Spotify API can be used to obtain audio metadata that contain more information than the local functions currently can return. Spotify audio features objects, obtained using the API, contain several variables to describe the song: duration in milliseconds, key, mode (major or minor) loudness (in decibel), tempo (beats per minute), time_signature (number of beats in each bar), energy (a perceptual measure of intensity and activity), "danceability" (how suitable a track is for dancing based on a combination of elements such as tempo, rhythm stability and beat strength), "instrumentalness" (prediction of whether a track contains no vocals), "acousticness" (a confidence measure of whether the track is acoustic or not), "liveness" (detection of the presence of an audience), "speechiness" (detection of spoken words), and valence (how positive/glad or negative/sad the track sounds).

Various image parameters, obtained either in the preprocessing phase (i.e., while generating the image) or at the postprocessing phase (image analysis of the finished image) can be used to characterize the images. Image metadata is implemented as an enumeration of tool parameters. This enumeration tells the panting algorithm how to parameterize each painting tool. This set of parametrized tools gives a direct description of how the resulting image will look. However, for another type of image analysis, such as pattern recognition or search for other hidden image features, postprocessing is required. One such postprocessing function has been implemented, which extracts a colour palette from the image using the k-means clustering algorithm. The returned colour palette can be closer to the perceived colours in the image than the original palette used due to colour mixing during painting. Some evolutionary art projects use image analysis for evolution (Machado & Cardoso, 1998; Klinger & Salingaros, 2000). The present framework contains functionality to evolve raw images. This means that the phenotypes in the EA are images, and that the fitness functions directly analyze the images. The system uses evolution on metadata, and analyzes the parameters used to create the end result rather than analyzing the end result itself.

Formally, a **mapping table** $t \in T$ (where T is the set of mapping tables that the painting algorithm utilizes) is a function used for feature mapping, utilizing image parameters as keys (K) and music parameters as values (V): $t = f : K \to V$. An image r is created by adding functions of music parameters. Such a function of the music parameters can be denoted as a painting tool. "The Image Artist" uses several tools ($f_1 \dots f_k$) to create an image r, each of them being a function of the music parameters

$$r = f_1\left(p_1^m, ..., p_i^m, ..., p_N^m\right) + \dots + f_k\left(p_1^m, ..., p_i^m, ..., p_N^m\right)$$

where $m = \left(p_1^m, ..., p_i^m, ..., p_N^m\right) \in M$ define the music parameters of a music file m belonging the set M of music files on which the painting algorithm operates.

The purpose of the **genotype** is to create a recipe for the painting algorithm that describes how music feature values are mapped into painting tool parameter values, e.g., musical tempo can be mapped to amount of brush strokes to paint, so that slow melodies create calm images, while high tempo melodies create chaotic images using lots of strokes. Hence the genotype can contain a mapping between the image feature 'brush strokes' and the music feature 'tempo' with a scaling interval $[20, 300]$, or a mapping between 'base colour' and music 'energy' with an interval $[270, 0]$.

The **phenotype** is an artist object ("The Image Artist"), which utilizes a mapping table representing the genotype. The task of the phenotype is to create an image recipe that can be used to paint the final image. For each tool parameter (key in the mapping table), the associated value is fetched. The value of the music parameter is used to calculate an output tool parameter value by linearly scaling the music value to the output interval.

Phenotypes are evaluated by a **fitness function** which uses subfunctions that estimate various image criteria. The number of subfunctions depends on the evaluation criteria and given goals of the image creation process. Examples of criteria can be user-defined aesthetical fitness, novelty function, and their combination. Optimum fitness is reached when the distance between the current genotype and an optimum mapping table is zero. As detailed below, three fitness functions have been implemented: optimizing towards a user specified mapping table, novelty combined with a user specified mapping table, and optimizing towards user preference without knowing the mapping table.

The "Optimizing towards a user specified mapping table" fitness function guides the evolution to find a mapping table that is "close" to what the user has specified. The distance between any two mapping tables is the sum of distances between key-value pairs in the mapping tables. Each tool parameter (key) should map to the correct music feature variable (value) and have the correct output interval. The Image Artist uses the output interval to calculate a value for a painting tool parameter. Given a target interval $T = [t_1, t_2]$ and current interval $C = [c_1, c_2]$, the distance between the intervals is $d\,(T, C) = |t_1 - c_1| + |t_2 - c_2|$. For mismatching music variables ($m_1 \neq m_2$) for a given tool parameter, the distance between the intervals is multiplied by a penalty factor k. The fitness function for the current genotype G is then calculated as a sum of contributions from all tools:

$$f\,(G) = \sum_{i=TP_1}^{N_{TP}} (d(T, C)k(m_1, m_2))_i$$

where N_{TP} is the number of tool parameters.

The fitness function "Novelty combined with user specified mapping table" creates a notion of novelty in the evolution, by optimizing towards a user suggested mapping table but ignoring some of the map entries and letting the system stochastically select how those tool parameters are mapped to music parameters and output intervals. Stochastical selection of parameter values is done through the nature of EA by not calculating fitness values for some parameters, therefore allowing any values for these parameters to propagate in the evolution. The EA uses the same distance and penalty functions as the previous fitness function, but does not iterate over all possible tool parameters: Some arbitrary tool parameters are not included in the fitness calculation, so some differences between the user selected mapping and the generated one will not be calculated in the fitness value. This fitness function can ignore *(a)* whole entries based on a key (tool parameter), *(b)* one mismatch between tool and music parameters, or *(c)* differences between output intervals. Hence user suggestions (representing user's aesthetic criteria) and novelty can be combined, and hopefully provide certain aesthetic qualities. The user can flag parts of the mapping table that the system can explore within. Ignoring user specifications leaves the system to arbitrarily select variables and values to use, giving it the possibility to introduce novelty in the results.

"Optimizing towards user preference without knowing the mapping table" operates somewhat differently compared to the previous two fitness functions. Instead of having an optimum mapping table to optimize towards, this fitness function is guided by user descriptions of how the final result should be. The fitness function analyzes how the mapping table affects the image in each genotype and compares these results with the provided information for each song. The user can, e.g., specify how the final colour palette should be without stating how the palette should be generated. Thus, the system will be missing information about critical parameters, and must find a mapping table that can generate the requested final result. This guides the evolution to search for mapping tables that match requested end results rather than predetermined mapping tables, so that the evolutionary algorithms can introduce interesting and unexpected mapping tables. The fitness function leverages a distance measure $d\,(A, B) = |A - B|$, where A is the user requested result and B the currently generated result. Depending on the opinion of the user, A and B can have different meanings, varying from colour palettes to the total number of brush strokes. This fitness function can introduce unexpected mapping tables that match user preferences but have interesting effects with other music. It can also be combined with one of the previous mentioned fitness functions, such that concrete user preferences can be combined with abstract preferences.

Furthermore, the system allows for varying the evolutionary algorithm's selection strategies, crossover techniques and mutation. Two **selection** strategies and two **crossover** alternatives have been implemented, proportionate selection and tournament selection resp. One-Point crossover and Uniform crossover (see, e.g., Floreano and Mattiussi, 2008). One-point crossover slices two mapping tables at an arbitrary index and combines the two parts from each genotype into a new genotype. Uniform crossover iterates over all keys and stochastically selects which value from the two tables to duplicate into the new genotype. **Mutation** is an essential part of the evolution, which is necessary to introduce diversity among the population and ensure a more complete search in the domain space. Mutation techniques can be modelled as stochastic processes that influence offspring. Having a mapping table as genotype, a new mutation technique must be implemented such that all parts of the table are mutable. This means that key-value pairs can be altered, and the information within the values can be modified. With a given probability, the feature variable is altered, such that selected tool parameter (key) is mapped to a different feature variable, or the output range is altered using a given mutation pressure. The mutation pressure in an interval $[-t, t]$ from which a random value in this interval is selected and added to a numeric variable selected for mutation.

The **painting algorithm** as such is mainly for creating abstract art by using different shapes and simulated brush strokes. However, it is not limited to abstract art: having simulated brush strokes allow for the creation of many art styles. A range of painting tools (for shapes, brush strokes and image effects) can be combined using layers, where each tool creates a layer on top of a digital canvas. The tools are highly parameterized to utilize each to its full potential.

Three main **geometric shapes** have been implemented: rectangles, ovals, and polygons. Rectangles can segment an image into sections or represent some objects. A combination of multiple rectangles in specific positions on the canvas can be used to create representations of more complex structures. Ovals are also useful for the representing objects, but since ovals have no edges, they can be used to enhance a calm emotion, a smooth motion or a soft object. Polygons are painted using random colours biased towards red. The number of points to be used, the positions of them, thickness and border colours are all parameters that can be set. Having many random points often yields pointy objects that can be related to aggressiveness and anger.

Two types of **brush strokes** are implemented: curved and straight. Curved strokes try to simulate brushes with a circular head, while straight strokes simulate brushes with a flat rectangular head. Simulating brush strokes can help the created images look creative, since humans can relate to results from human artists. Both types of brush strokes are implemented using a high number of regular straight lines, all following the same direction (from start of the stroke to the end). Every line within a stroke is altered differently as the stroke is painted to give the effect of paint smudging and fading. In the straight brush strokes, the colour intensity degrades as the stroke is painted, and fades out at the end. In the curved strokes, this effect is slightly reduced as it naturally occurs due to the layout of lines. The curved brush has all its painting lines in a 2D normal distribution, while the straight brush has lines evenly spread out among its width.

Currently there are three types of **image effects** implemented: cloud, blur and oil. The cloud effect mostly affect the colours of an object by creating a monochrome layer of noise that can induce some diversity among equal shapes. The blur effect softly smooths out sharp corners in an image, while the oil effect removes some of the clearly artificial lines resulting from the brush painting algorithm, so that the resulting image reminds of an artistic effect involving water or oil, rather than computer generated curved lines.

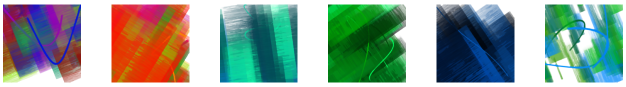
4 Experiments and Results

Various experiments were performed to validate system behaviour and to explore the importance and sensitivity of various techniques and approaches. Table 1 shows a basic experimental configuration. The population size is limited due to limited computational resources and time. A set of six songs was selected for these experiments. They differ from each other on several musical features, but are also equal in some features: *Billie Jean*, Michael Jackson; *Chained To The Rhythm*, Katy Perry, *I Promise*, Alex Kozobolis; *Kaleidoscope*, Kasbo; *No Time For Caution*, Hans Zimmer; and *The Imperial March (Darth Vader's Theme)*, John Williams.

Table 1: Basic EA configuration for the evolutionary runs

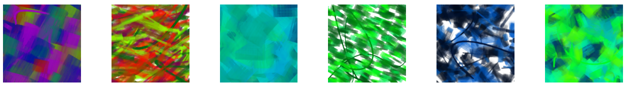
EA option	Value
Population size	20
Max generations	2000
Elites	1
Crossover rate	0.7
Mutation rate	0.7
Parent selection	Tournament
Crossover type	One Point
Mutation pressure	20

Billie Jean	Chained To The Rhythm	I Promise	Kaleidoscope	No Time For Caution	The Imperial March



(a) Optimisation on user preferences

(b) Novelty search freely selecting colour variables



(c) Novelty search ignoring some user-defined keys

Figure 2: Evolution towards a user-defined mapping table

Experiments with fitness functions

Experiments with different fitness functions were performed to investigate how they influence end results and whether some parts of the function can guide the EA to fulfill some objectives. A small change in the fitness function can introduce novelty in the results or guide the EA directly towards user preference. The fitness evaluation can be based on the mapping table or the painted images.

The first experiments used the fitness function “Optimizing towards user specified mapping table”, by first evolving a mapping close to the user’s suggestion. Figure 2a shows the generated images after evolution. The colour palette is generated using the music feature variables energy and valence. “Chained To The Rhythm” has high energy and yielded a colour palette based on the colour red, while “I Promise” has low energy and got a colour palette in the blue spectrum, which fitted well with what the user had specified.

Figure 2b shows images generated by evolving towards a user preferred mapping table, but ignoring user specified parameters for colour, allowing the system to freely select the colour variables. Comparing Figure 2a and Figure 2b, the major difference is in the colour palette in each image. The “Billie Jean” image in Figure 2b has multiple bright colours, while in Figure 2a the colour palette is darker; however, the user claimed that both colour palettes fitted the music of “Billie Jean”. For “The Imperial March”, the user thought the image in Figure 2b fitted the music better than the one in Figure 2a, due to the presence of dark and red-pink colours. This was surprising and appreciated by the user.

A third experiment ignored some arbitrary parameters in the user preferred mapping table, to see whether the resulting images can surprise the user, while matching most of their preferences. Another purpose was to see how sensitive the system is to the parameters. The generated images introduce style variance by mostly differing from the previous ones in shape construction, with Figure 2c using a high number of small shapes, while Figure 2a uses few big shapes. It is visible that the system is sensitive to changes in parameter values. The user agreed that the images in Figure 2c fit to the music, but also introduce a positive element of surprise.

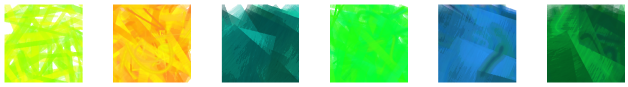
Billie Jean	Chained To The Rhythm	I Promise	Kaleidoscope	No Time For Caution	The Imperial March

(a) Base colour preference of the user

(b) Optimisation on user preferred colours

(c) Optimisation on user preferred base, spread and figures

(d) Another optimisation on user preferred base, spread and figures



(e) Optimisation towards user preferred base, spread and figures, with seeding genotypes taken from earlier experiments (Figure 2a)

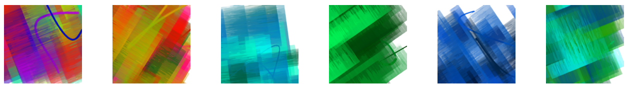
Figure 3: Evolution without a known mapping table

The set of experiments shown in Figure 3 optimize towards abstract user preferences without knowing the mapping table, meaning that the user specified how parts of the end result should be, rather than how to generate them. This evolution did therefore not have a known mapping table to optimize towards, but had to search for a mapping table fitting the user preferences. These preferences were extracted from a user survey, where for each song a base colour is selected, as well as image aggressiveness, and amount of brush strokes to use. The following experiments are based on the preferences from one arbitrary user, displayed in Figure 3a.

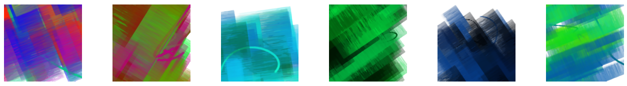
Figure 3b shows the results after evolution optimizing towards user preferred base colour for each song, and the colour spread in the palette. As there the amount of brush strokes is not optimised, the final amount happened to be high, so filling the whole painting canvas. Comparing the results to the user’s preferences, there are some differences in shades, but there is agreement on the base colours.

Figure 3c shows the results after evolution optimizing towards the user preferred base colour for each song, and the colour spread in the palette. However, this set was generated using all the available painting tools, to generate a set of images that differ from other experiments. Comparing Figures 3b and 3c, there are two different painting styles in the images. The images in Figure 3c are more dynamic, with the use of several painting tools. The polygons provide some aggressiveness to the images, while the small ovals give elements of surprise that contribute to novelty. Figure 3d is another set generated the same way as Figure 3c, but with different parameters. This image set is slightly more dynamic with the use of rotation in some brush strokes.

Billie Jean | Chained To The Rhythm | I Promise | Kaleidoscope | No Time For Caution | The Imperial March

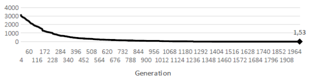


(a) Optimisation on user preference without seeding

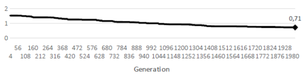


Billie Jean | Chained To The Rhythm | I Promise | Kaleidoscope | No Time For Caution | The Imperial March

(b) Optimisation on user preference with seeding



(c) Mean fitness without seeding



(d) Mean fitness with seeding

Figure 4: Experiments with seeding techniques

Figure 3e shows a result set after seeded evolution. The fitness function operates on preferences from the same user as in Figure 3b, but the evolution is seeded with genotypes from the experiment that produced Figure 2a. These two users have different preferences. Comparing Figures 2a, 3b and 3e shows similarities in all results, where Figure 3e shares image features from both experiments. This shows that seeding affects the images, and can introduce novelty.

The experiments in Figure 2) and Figure 3 used data from different users with different preferences, so the final results cannot be directly compared. However, all experiments did match user preference either through direct mapping tables or through specific requirements within the end results such as colours.

Experiments with seeding techniques

Seeding techniques are used to influence the initial population of the EA, to affect where in the search space the EA should start. Seeding has been met by scepticism (with, e.g., Eiben & Smith, 2015, claiming that it might be unnecessary), but can give the EA a push in the right direction Seeding can also be used to incorporate previously generated results matching user criteria, so that the EA can explore a local search space. These experiments aimed to investigate how seeding influences the results and EA performance, by (i) seeding initial population based on given genotypes, so that the search starts in a predetermined place in space, and (ii) initialize with fully stochastic population (no seeding).

All experiments were run five times, with the results averaged, and performed to observe how seeding affects both the evolution of mapping tables and resulting images. Evolution was optimized towards user specified mapping tables. Figure 4c shows the mean best fitness values without seeding. The steepest decline in fitness happens in the first 1000 generations. This experiment optimizes towards a user preferred mapping table, so it is expected to get results similar to Figure 2a. Figure 4a shows that this is indeed the case: the colours used are similar, as is the rotation of figures and amount of brush strokes. The visual differences are due to stochastic painting order and colour selection.

Figure 4d indicates that seeding drastically improves the performance of the EA in the first generations, as rediscovery of previous genotypes with low fitness values is avoided. However, after $2,000$ generations only slight improvements are noticable. Comparing the resulting images with (Figure 4b) and without (Figure 4a) seeding, it is visible that they share the same features. Figures 4c and 4d highlight the last best fitness value in each experiment, showing a mean improvement of only 0.82 across the two experiments (1.53 resp. 0.71). As Eiben & Smith (2015) stated, seeding is not necessary. The EA will eventually reach its target fitness value if configured correctly. However, if the objective is to reduce execution time, seeding can be an efficient option.

Experiments with user involvement

As the aesthetic judgement is subjective it is necessary to involve humans in the learning and evaluation process. Here, users were involved through a small survey and through interviews with one or two persons. The interviews were used to generate mapping tables the system can optimize towards. The questions were about relations between tool parameters and music variables, as well as numeric intervals. The survey was created to obtain a more general overview of user expectations on how the images should look, and to get feedback on the overall aesthetics of the generated images and how well they match the input music.

The user feedback presented below is a combination of individual responses and a summary of all users' responses. User expectations of how an image should look after listening to a specific song were described by four categories:

1. **Base colour** for palette generation, taking the user's response colour and making it darker or lighter if requested.
2. **Energy:** A measure from 1–5 where 1 is relaxing and 5 is aggressive. This scale is used to get an indication of how figures and brush strokes should be placed in the image.
3. The number of **brush strokes** to use in the image, on a scale from 1–10, where 1 is very few and 10 is many.
4. Expected colouring where the used **palette** should have:
   O1: One colour with small changes in shades
   F1: Few adjacent colours with small changes in shades
   M1: Many adjacent colours with small changes in shades
   O2: One colour with high variance in shades
   F2: Few adjacent colours with high variance in shades
   M2: Many adjacent colours with high variance in shades

To exemplify, Table 2 shows the user expectations for *Billie Jean*, *No Time For Caution* and *The Imperial March*, while Figure 5 summarizes the user feedback on the actual produced images for these songs (shown in Figure 2a). Most positive comments on *Billie Jean* related to the colours (the palette, the repetion and the relationship between colours), the amount of brush strokes, and that the image follows the rhythm. The negative comments included that it was too uniform, dark and geometrical, and had too many colors and exposed canvas. The image generated for *No Time For Caution* was mainly liked by the users, with comments that it was aesthetically pleasing, reflected the mood of the music, and had the right colours and colour temperature. The few negative responses said it was a bit too dark and needed more aggressive colours. On the other hand, *The Imperial*

Table 2: Individual user expectations

(a) Billie Jean

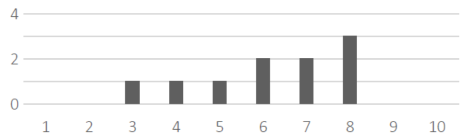
User	1	2	3	4	5	6	7	8	9	10
Base										
Energy	3	2	4	2	4	3	4	3	4	4
Strokes	7	3	8	5	9	5	8	7	7	8
Palette	M1	F1	M2	F1	F2	F1	F2	M1	F2	M1

(b) No Time For Caution

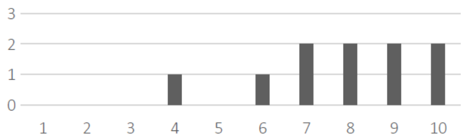
Base										
Energy	2	3	5	3	2	3	5	1	4	5
Strokes	1	6	9	7	2	5	9	2	8	6
Palette	O1	F1	F2	F2	F1	O1	F2	F1	F2	F2

(c) The Imperial March

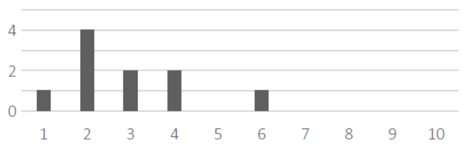
Base										
Energy	5	4	5	4	4	4	5	4	5	5
Strokes	10	8	10	4	6	6	9	8	10	8
Palette	F2	M1	F1	F1	F1	M1	M2	F1	M2	F2



(a) Billie Jean



(b) No Time For Caution



(c) The Imperial March

Figure 5: User feedback summary

March image was mainly disliked by users, who thought it had the wrong colours and mood, and was not aggressive enough. Still, the image got positive feedback on its small brush strokes, and for being dynamic and creative.

5 Discussion and Conclusion

This paper explores computer-generated art based on musical input, focusing on the use of evolutionary algorithms in image generation. The end-user involvement, survey results and user interaction tests, contributed to the system design of the evolutionary algorithm functionality, e.g., seeding techniques, feedback functions, and mappings between genotype and phenotype information. The evolutionary algorithms for music-to-image transformation operate on music and image meta-data, rather than raw data (music and image media). This requires a good metadata structure and organization, as well as good solutions for metadata and parametric representation of music and images. Some of the design achievements that should be emphasized are: *(a)* metadata-driven design of the genotype, *(b)* metadata-driven "coupling" between the EA and the painting tools, and *(c)* generic design of the fitness function (that gives a possibility to experiment with various fitness evaluation approaches).

As aesthetic judgement is subjective, it is difficult to create an automated evaluation process. Experiments with fitness functions and user involvement showed that the system was able to find a mapping table that is very close to the user preference, but that the resulting images sometimes were not optimal considering user expectations. The most important design tests and experiments focus on *(a)* using mapping tables to match music and image features and *(b)* various fitness functions that combine user preferences with art novelty aspects. The obtained experimental results and the end user interaction tests and evaluations are promising and motivate for further development of the tool set, as well as fine-tuning of the evolutionary algorithms.

The system can partially learn about user preferences through earlier experiences. The best genotypes after each evolution can be stored and reused through seeding. The key elements of one or more user preferences can then be collected through data analysis. The system can accurately reuse (single) user preference data through the seeding. The experiments show that the results may come from the evolution optimizing directly towards user preference, but also from fitness functions optimizing for novelty. The novel results can be approved by the user and included in the experience data. Diverse user preferences make it difficult to generate a mapping table fitting every user's preferences, so currently the system can at best learn individual user preferences. So far, not enough data analysis modules have been implemented to take full advantage of earlier experiences. Intelligently merging previously generated genotypes (based on how different music and image parameters affect end results and sensitivity) could produce more accurate solutions.

Assuming that the provided user preference through the metadata is accurate enough, the system can create images that are both æsthetic and meaningful. In some cases, there was some negative feedback on the image æsthetics and the music match. This is mainly due to subjective preferences of image quality. Even though users are not directly involved in the evolution, some user interaction can be introduced through the seeding. The seeding could be introduced mid-evolution to push the evolution in a specific direction (by choosing the genotypes with specific/wanted properties).

Due to subjective judgement of æsthetics and cultural differences, the system cannot create æsthetically pleasing and meaningful images without any user involvement, as also pointed out by Galanter (2010). This system thus involves end users in the initial stage of the evolutionary algorithms to obtain some guidelines towards user preferences. The system is able to generate pleasing images for end users that share at least some notions of aesthetics, such that the differences between the user's preferences can be utilized positively. The system can generate images based on one user's preferences and thus be considered as novelty by another user. In this scenario, the second user has no involvement in the system. However, there is no implemented fitness function that is able to cover every user's preferences.

The experiments shown in Figure 2 confirm that the parameters used for the fitness function influence the style of the results, as noted by den Heijer & Eiben (2010). They pointed out that this might not be beneficial for the application. However, our analysis shows that for some use cases it might be beneficial, e.g., for "Dynamic Ambient Decoration" and "Therapeutic Art", while other use cases might require more novelty and artistic freedom, e.g.,"Artist's Work Tool" and "AI Art Generator". where the computer should be able to generate high quality and novel art, either through interplay with and guidance from an artist or completely self-sustained.

The framework enables improvements in several directions. For instance, evolutionary algorithm improvements (different genotype, phenotype, mutation and fitness function solutions), alternative approaches to music-to-image transformation, utilizing additional music and image features to enrich the results, interfacing other music and image systems and platforms and using the additional information and knowledge they can offer, and interacting with the end-user in new ways, e.g., creating web platforms that can learn by interaction with user groups (inspired by Trujillo et al., 2013 and García-Valdez et al., 2013).

References

Ashmore, L. & Miller, J., (2004). Evolutionary Art with Cartesian Genetic Programming. *Technical Online Report*.

Baluja, S., Pomerleau, D. & Jochem, T., (1994). Towards automated artificial evolution for computer-generated images. *Connection Science*, 6:325–354.

Bredesen, T.H., Carlsen, J., Seem Koren, H.K., Serussi, S. & Strømjordet, E. (2016). *Sågawave, Creating visualizations based on attributes in music*. Trondheim: NTNU.

Clune, J. & Lipson, H., 2011. Evolving three-dimensional objects with a generative encoding inspired by developmental biology. In *ECAL*., 2011.

Cohen, H., 1988. How to Draw Three People in a Botanical Garden. In *AAAI*., 1988.

Colton, S., (2012). The Painting Fool: Stories from Building an Automated Painter. In J. McCormack & M. d'Inverno, eds. *Computers and Creativity*. New York: Springer. pp.3–36.

den Heijer, E. & Eiben, A.E., (2010). Comparing Aesthetic Measures for Evolutionary Art. In *Applications of Evolutionary Computation: EvoApplications 2010, Proceedings, Part II*. Springer. pp.311–320.

Dumoulin, V., Shlens, J. & Kudlur, M., (2016). A Learned Representation For Artistic Style. *arXiv*, Available at: https://arxiv.org/abs/1610.07629.

Eiben, A.E. & Smith, J.E., (2015). Interactive evolutionary algorithms. In *Introduction to Evolutionary Computing*. Berlin Heidelberg: Springer. pp.215–222.

Galanter, P., (2010). The Problem with Evolutionary Art Is. In *Applications of Evolutionary Computation: EvoApplications 2010, Proceedings, Part II*. Springer. pp.321–330.

Galvin-López, E., McDermott, J., O'Neill, M. & Brabazon, A., (2011). Defining locality as a problem difficulty measure in genetic programming. *Genetic Programming and Evolvable Machines*, 12:365–401.

García-Valdez, M., Trujillo, L., de Vega, F.F., Guervós, J.J.M., & Olague, G. 2013. Evospace-interactive: A framework to develop distributed collaborative-interactive evolutionary algorithms for artistic design. In *International Conference on Evolutionary and Biologically Inspired Music and Art*., 2013.

Johnson, C.G., (2016). Fitness in evolutionary art and music: a taxonomy and future prospects. *International Journal of Arts and Technology*, 9:4–25.

Klinger, A. & Salingaros, N.A., (2000). A pattern measure. *Environment and Planning B: Planning and Design*, 27:537–47.

Kowaliw, T., Dorin, A. & McCormack, J., 2009. An empirical exploration of a definition of creative novelty for generative art. In *Australian Conference on Artificial Life*, 2009.

Lehman, J. & Stanley, K.O., 2010. Revising the evolutionary computation abstraction: minimal criteria novelty search. In *Proceedings of the 12th annual conference on Genetic and evolutionary computation*., 2010.

Lewis, M., (2008). Evolutionary visual art and design. In *The art of artificial evolution*. Springer. pp.3–37.

Machado, P. & Cardoso, A., 1998. Computing aesthetics. In *Brazilian Symposium on Artificial Intelligence*., 1998.

Machado, P. & Cardoso, A., (2002). *All the truth about NEvAr*. Portugal: Applied Intelligence 16(2):101–119

Re, A., Castelli, M. & Vanneschi, L., 2016. A Comparison Between Representations for Evolving Images. In *International Conference on Evolutionary and Biologically Inspired Music and Art*., 2016.

Secretan, J., Beato, N., D'Ambrosio, D.B., Rodriguez, A., Campbell, A., Folsom-Kovarik, J.T., & Stanley, K.O (2011). Picbreeder: A case study in collaborative evolutionary exploration of design space. *Evolutionary Computation*, 19:373–403.

Sims, K., (1991). *Artificial evolution for computer graphics*. ACM.

Todd, S. & Latham, W., (1992). *Evolutionary art and computers*. London: Academic Press, Inc.

Trujillo, L., García-Valdez, M., Fernández-de-Vega, F. & Merelo, J.-J., 2013. Fireworks: Evolutionary art project based on evospace-interactive. In *Evolutionary Computation (CEC), 2013 IEEE Congress on*., 2013.

Vinhas, A., Assunção, F., Correia, J., Ekárt, A., & Machado, P., 2016. Fitness and novelty in evolutionary art. In *International Conference on Evolutionary and Biologically Inspired Music and Art*, 2016.

Woolley, B.G. & Stanley, K.O., 2011. On the deleterious effects of a priori objectives on evolution and representation. In *Proceedings of the 13th annual conference on Genetic and evolutionary computation*., 2011.

MIDI Database and Representation Manager for Deep Learning

Jeff Ens, Philippe Pasquier
School of Interactive Arts and Sciences
Simon Fraser University
{jeffe,pasquier}@sfu.ca

Introduction

One of the most important decisions made when training neural networks, is how to represent the data. Despite the large number of possible representations, the piano roll dominates recent literature on modelling music [2, 3]. Furthermore, previous work suggests that modelling simpler conditional probability distributions such as $p(\text{pitch}|\text{rhythm})$, may be an advantageous approach to the complex task of modelling music [5]. However, many alternate representations are more difficult to implement than the piano roll. Motivated by these factors, we propose an accessible framework for symbolic musical data storage, and dataset construction, which supports a variety of representations.

Representations

Base Representation

Information extracted from a MIDI file is stored in a .tfrecords file, to be compatible with the Tensorflow [1] Dataset API. We store the start time, end time, pitch, velocity, channel number, and midi instrument of each note. In addition, we store the start time, and length of each measure, which is useful when a representation involves separating a piece into measures. All time-based attributes are stored in ticks, which allows the user to specify the degree of quantization at run-time, and preserves the original MIDI data without loss of information. Notablye, additional metadata such as the title, composer, and bpm is also stored.

Univariate Representations

A univariate representation is a sequence of one-hot vectors, encoding a sequence of $\langle$state, value$\rangle$ tuples. We represent each *state* and all valid *state-transitions* using a directed graph, as shown in Figure 1. Figure 1 shows a generalized version of the representation proposed by Liang and others, where the pitches sounding in each timestep are delimited by the step state [4]. Additionally, Liang and others specifies that the pitches in each timestep must be arranged in ascending order. To accomodate these types of representations, we provide a simple interface for specifying *states*, valid *state-transitions* and constraints on the value of a state dependant on the values of previous states. It is very straightforward to construct more complex representations. Once a directed graph has been specified, the conversion between a $\langle$state, value$\rangle$ tuple and a one-hot vector is provided via a single function call. Furthermore, sequences can be validated, ensuring that there are no invalid *state-transitions* and that no constraints are violated.

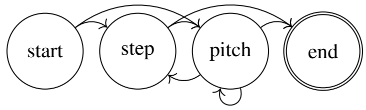


Figure 1: A univariate representation.

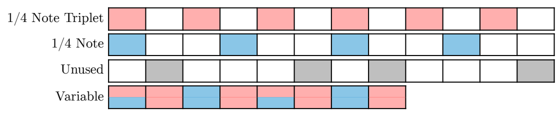


Figure 2: Fixed and variable resolution representation.

Multivariate Representations

A multivariate representation is a sequence of binary vectors (stored as a matrix). The most common multivariate representation is the piano roll. Typically, piano rolls are constructed using a fixed resolution (i.e. 16th note resolution), however, we also support a variable resolution representation. The benefits of using a variable resolution are shown in Figure 2. Consider a rhythm consisting of $1/4$ note triplets and standard $1/4$ notes with the length of a single beat. A fixed resolution representation would require 12 bits to represent the rhythm, however, 4 of these bits would never be used. The variable representation would only require 8 bits.

References

[1] Abadi, M., et al. 2016. Tensorflow: Large-scale machine learning on heterogeneous distributed systems.

[2] Boulanger-Lewandowski, N., et al. 2012. Modeling temporal dependencies in high-dimensional sequences: Application to polyphonic music generation and transcription. In *Proc. of the 29th Int. Conf. on Machine Learning*, 1159–1166.

[3] Huang, C. A., et al. 2017. Counterpoint by convolution. In *Int. Conf. on Learning Representations (under review)*.

[4] Liang, F., et al. 2017. Automatic stylistic composition of bach chorales with deep lstm. In *Proc. of the 18th Int. Symposium for Music Information Retrieval*, 449–456.

[5] Walder, C. 2016. Modelling symbolic music: Beyond the piano roll. In *Proc. of the 8th Asian Conference on Machine Learning*, 174–189.

Integrating ETHNO-MUSIC and Tra-la-Lyrics for Composing Spanish Popular Songs

Maria Navarro[1] and **Hugo Gonçalo Oliveira**[2]
[1] BISITE Research Group, University of Salamanca, Spain. maria90@usal.es
[2] CISUC, Department of Informatics Engineering, University of Coimbra, Portugal. hroliv@dei.uc.pt

Overview

An automated composer of popular Spanish songs was developed by integrating ETHNO-MUSIC (Navarro-Cáceres, Olarte, and Cardoso (2018)), for generating melodies, and Tra-la-Lyrics (Gonçalo Oliveira (2015)), for producing lyrics. This confirms that it is not always necessary to develop new creative systems from scratch.

Spanish popular music differs from classical in aspects like sonority, sound disposition or rhythmic formulas used. It is always linked to a functionality and lyrics are essential for identifying the song's purpose.

Presented integration is analogous to having two different people composing a song: one creates the melody and another the lyrics. Unlike other systems that generate music for existing lyrics (e.g. Toivanen, Toivonen, and Valitutti (2013); Ackerman and Loker (2017)), in this case, melody is composed first,

ETHNO-MUSIC generates new melodies based on original Spanish popular songs, available in MIDI format. Musical excerpts were analyzed and their relevant features (pitch, duration, degree, first in bar, time signature, musical phrase) encoded and stored to be used as the training corpus for music. A Markov Model (MM), learned on the previous features, is then used for generating new compositions.

Once the melody is available, Tra-la-Lyrics 2.0, built on top of the poetry generation platform PoeTryMe, splits the melody into parts and generates lines of text for each part, while trying to maximize two main constraints: (i) one syllable per note; (ii) stressed syllables match strong beats of the melody. Generation is based on the Spanish adaptation of PoeTryMe (Gonçalo Oliveira et al. (2017)), though with an augmented semantic network, acquired from ConceptNet, and new line templates, acquired automatically from songs in Spanish.

Example

The MM was learned from 102 popular songs with time signature 3/4, 3-4 musical phrases with similar length, and similar sonority in the Frigian mode with possible modifications in its evolution to E minor.

Lyrics were produced for a set of melodies generated with ETHNO-MUSIC, with seed words that set two generation domains, common in Spanish popular songs: work in the fields (*trabajo, siega, tierra, sembrar, semillas, trigo, cereales, campo, sol, paja, cosecha, cosechar*); and love (*amor, novia, moza, mozo, bella, belleza, feliz, alcoba, morena, guapa, sonrisa, ojos, bonito, bonita*). Figure 1 has an example of a song generated in the domain of love, with lyrics roughly translated to English. Most stressed syllables match strong beats and rhymes are frequent.

You look your appearance in the shape of vulgarity
because of a wonderful and irresistible disease
Your chin and face, my sweet love / somebody stole the key of my heart

Figure 1: Song with lyrics about 'amor' (love), including rough English translation.

A set of generated songs was evaluated by human judges. Overall results suggested that melodies transmit a feeling of Spanish popular music and both the rhythm and meaning of the lyrics is acceptable for a first approach.

References

Ackerman, M., and Loker, D. 2017. Algorithmic songwriting with ALYSIA. In *Proceedings of 6th International Conference on Computational Intelligence in Music, Sound, Art and Design*, EvoMUSART, 1–16.

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

Gonçalo Oliveira, H. 2015. Tra-la-lyrics 2.0: Automatic generation of song lyrics on a semantic domain. *Journal of Artificial General Intelligence* 6(1):87–110. Special Issue: Computational Creativity, Concept Invention, and General Intelligence.

Navarro-Cáceres, M.; Olarte, M.; and Cardoso, A. 2018. User-guided system to generate Spanish popular music. In *Proceedings of 9th International Symposium on Ambient Intelligence*. Springer Verlag.

Toivanen, J. M.; Toivonen, H.; and Valitutti, A. 2013. Automatical composition of lyrical songs. In *Proceedings of 4th International Conference on Computational Creativity*, ICCC 2013, 87–91. Sydney, Australia: The University of Sydney.

Generation of Aesthetic Emotions guided by Perceptual Features

Ana Rodrigues, Amílcar Cardoso, Penousal Machado
CISUC, University of Coimbra
Coimbra, Portugal
(anatr, amilcar, machado)@dei.uc.pt

Abstract

We present an experimental prototype that aims the study of aesthetics-related features from visual and auditory domains to express a set of 13 emotions. In the visual domain features are unfolded with a chance of occurring according to their perceptual relevance, whereas in the auditory domain there is a previous categorization of emotions. In the end this will result in a series of digital abstract faces expressing certain emotional states.

Introduction

Understanding how to evoke a certain emotion through sound (Juslin, 2013) and image (Lindborg and Friberg, 2015) is a crucial point for the development of design artefacts based on non-verbal communication. Nevertheless, this issue is still largely unexplored by design sciences.

Based on several experiments previously conducted by multiple authors, we argue that the expression and communication between these seemingly distinct domains is easier and more comprehensible through the development of a perceptually relevant aesthetical language.

The modularity provided by digital frameworks allows the creation of dynamic environments in quick and efficient ways. As such, we propose to develop this language which we then incorporate into a computational prototype to express a specific set of aesthetical emotions.

The Prototype

It is known that several visual aspects may influence the induction of emotions. For instance, brighter colors have been linked to positive emotions, whereas darker colors have been linked to negative emotions (Lindborg and Friberg, 2015). Other associations have been studied by Cavanaugh (Cavanaugh, MacInnis, and Weiss, 2016).

In our work every emotion has a corresponding visual and auditory expression. Visual features have weights (probability of occurring) according to emotion in question. The visual alphabet is then composed by high-level features (complexity, density, texture), low-level features (line, shape, size, color), and manipulations (motion, repetition, symmetry). Inspired by Chernoff faces (Chernoff, 1973) modularity and nature, we generate several digital faces with properties guided by our previously mentioned visual alphabet. On the other side, music was subject to a previous emotion categorization based on the piece character.

Conclusions and Future Work

We made use of the digital computer's capacity to transform and manipulate multidimensional data to investigate cross-modal associations in creative ways. This work aims at contributing to a better understanding of the foundations of associations between visual and auditory domains, by providing both scientific and aesthetical foundations to solve problems of cross-domains.

Although significant bibliographic research and experimentation has been done in this work, it must be subject to continuous updates in the future. We believe that the use of IEC (Interactive Evolutionary Computation) is important to evaluate the relevance of specific features regarding a specific emotion. Using this type of approach, the user will be able to redefine or reorganize the corresponding weight of each feature, allowing the system to learn with the user, interactively, how to evolve the audiovisual mappings. This can result in two benefits: the generation of mappings that suit the preferences of the user; and the analysis of the interactions that will allow the understanding of the user's perceptual motivations.

References

Cavanaugh, L. A., MacInnis, D. J., & Weiss, A. M. (2016). Perceptual dimensions differentiate emotions. *Cognition and Emotion*, *30*(8), 1430–1445.

Chernoff, H. (1973). The use of faces to represent points in k-dimensional space graphically. *Journal of the American Statistical Association*, *68*(342), 361–368. http://doi.org/10.1080/01621459.1973.10482434

Juslin, P. N. (2013). From everyday emotions to aesthetic emotions: Towards a unified theory of musical emotions. *Physics of Life Reviews*. http://doi.org/10.1016/j.plrev.2013.05.008

Lindborg, P., & Friberg, A. K. (2015). Colour association with music is mediated by emotion: evidence from an experiment using a CIE lab interface and interviews. *PloS One*, *10*(12), e0144013.

A Set of Procedures Attempting at the Generation of Verbal Humor in Portuguese

Hugo Gonçalo Oliveira[1] and **Ricardo Rodrigues**[2]
[1] CISUC, Department of Informatics Engineering, University of Coimbra, Portugal
[2] CISUC, Polytechnic Institute of Coimbra, Portugal
hroliv@dei.uc.pt, rmanuel@dei.uc.pt

Overview

Automatic generation of verbal humor has been tackled by different authors, following different approaches and rendered differently, generally in English. Inspired by them, we implemented a set of procedures for producing humoristic riddles or definitions, but in Portuguese, taking advantage of the linguistic resources available for this language. To the best of our knowledge, besides the automatic generation of Internet memes (Gonçalo Oliveira, Costa, and Pinto (2016)), this constitutes the first attempt for the automatic generation of humor in Portuguese.

Models and examples

Six models were implemented, based on the reinterpretation of known concepts or creation of new ones with a similar sound. Here, we briefly describe the models and, for each, present an example obtained by exploiting a knowledge base with semantic relations (Gonçalo Oliveira (2018)), used as features, and, when necessary, a list of compounds (Ramisch et al., 2016). Results can be rendered as a question-answer pair (riddle) or as a definition.

- **Reinterpretation of compounds:** given a known noun+adjective compound, features are acquired for each of its words, individually, and used to (re-)define it. As the meaning of the compounds is generally more than just the sum of the meanings of its words, this may result in unexpected associations, possibly perceived as incongruent, and thus humour-prone.
  - As a definition: *direitos humanos: um homem plano* (human rights: a plain man)
  - As a riddle: *Que resulta do cruzamento entre o que é plano e um homem? direitos humanos.* (What do you get when you cross a plain and a man? human rights)
- **New compounds:** as humour may result from words with similar sounds, new concepts are obtained with a single edition to the words in compounds. These can also be defined with features of their words.
  - *ponto forte* (strong point) → *porto forte: um vinho que é um lado* (strong port: a wine that is a side)
- **Reinterpretation of words:** similar to the reinterpretation of compounds, this time with single words, interpreted as a blend of two (portmanteau). This often leads to an unexpected and, to some extent, creative meaning, attributed to the word, again perceived as incongruent, thus increasing the humour potential.
  - *soldado* (soldier) = *sol+dado* (sun+data) → *soldado: uma luz que é concedida* (a given light)
- **New blends:** similar to the previous, but involving a minimal change in the words, though keeping a similar sound. It leads to a new concept with a sound that resembles a known one, interpreted as the blend of two other (often unrelated) concepts.
  - *divertido* (funny) → devertido = *dever+tido* (duty+had) → *devertido: um trabalho que é considerado* (a work considered).
- **Partial antonyms:** the orthography of some words starts or ends with the antonym of another. Novel antonyms may result from changing the start/end of those words with its antonym.
  - *bombeiro* (fireman) = *bom+beiro* (good+beiro) → *maubeiro* = *mau+beiro* (bad+beiro)
- **Blend of antonyms:** other words can be interpreted as the blend of other two words, each with its own antonym.
  - *friolento* (chilly) = *frio+lento* (cold+slow) → *calor-rápido* (fast-hot).

We are satisfied with some of the results produced, but not so much with others. Thus, we are currently testing ways for automatic ranking the produced results and we will soon conduct an evaluation by human subjects.

References

Gonçalo Oliveira, H.; Costa, D.; and Pinto, A. 2016. One does not simply produce funny memes! – explorations on the automatic generation of internet humor. In *Proc 7th Intl Conf on Computational Creativity*, ICCC 2016.

Gonçalo Oliveira, H. 2018. A survey on Portuguese lexical knowledge bases: Contents, comparison and combination. *Information* 9(2):34.

Ramisch, C.; Cordeiro, S.; Zilio, L.; Idiart, M.; and Villavicencio, A. 2016. How naked is the naked truth? a multilingual lexicon of nominal compound compositionality. In *Proc 54th Annual Meeting of the Association for Computational Linguistics (Short Papers)*, 156–161. ACL Press.

Computational Creative Experiments in the Development of Visual Identities.

Sérgio Rebelo, João Bicker and Penousal Machado
Department of Informatics Engineering of Faculty of Sciences and Technology
University of Coimbra, Portugal
{srebelo, bicker, machado}@dei.uc.pt

Abstract

The democratisation of creative tools has created an unstable professional environment and a degradation of Graphic Design artefacts. However, we believe that creative computational tools will have key importance in the future of the profession, allowing for the development of more replicable, scalable, and user-oriented work. Therefore, we propose a creative computational experiment to develop a visual identity generator that uses an Evolutionary Algorithm to simulate the traditional design process.

Introduction

Nowadays, the computer is the primary tool of production in Graphic Design (GD) and digital media design is one of the more attractive design fields [Blauvelt, 2011]. However, the democratisation of creative tools has lead to a floating and unstable professional environment, which has promoted the appearance of crowdsourcing platforms for creatives, such as 99Designs [Shaughnessy, 2012]. In these platforms, a user can submit a proposal and have "dozen of designers" work for them, albeit, generally, the outcomes appear to be no more than "template-based designs." We believe that the future of the profession will depend on creative computational tools. These tools will enable the creation of replicable, scalable and more audience-oriented work. This way, the graphic designers will become mediators between the tools and the clients [Armstrong et al., 2012]. With this in mind, we have developed an Evolutionary Algorithm (EA) that creates visual identities through a set of sensations defined by the user. The system replicates a process like creative crowdsourcing platforms where several "candidate proposals" are developed/generated and iteratively evaluated.

Approach

In his book *Designing Programmes* (1964), Karl Gerstner presents the design process as the simple act of picking out determining elements and combining them. On this principle, we have developed a proof-of-concept for a semi-autonomous visual identity designer/generator, which by means of an EA, can generate logotype designs. The system works in a semi-autonomous way wherein the user defines, a priori, a set of sensations that the outcome should transmit (e.g. age, luxury, complexity, etc.). The system consists of two main modules: (1) the "creative" that is responsible for the development of the logotypes; and (2) the "appraiser" that assess the designs generated by the "creative," according to the relationship between its elements and the data defined by the user.

Each logotype is represented by a shape, a lettering style, and a set of stylish elements. In this initial setup, we defined five shapes and six lettering styles that can be combined in two distinct positional styles. Aside from this, the logotype can to be adorned with different stylish elements (inner stroke styles and decorative elements). With this, the "creator" module is responsible for the initialization of a logotypes' population and, subsequently, for the implementation of the recombination and mutation operators. The "appraiser" module assess the fitness of the candidate solutions and selects the best outcomes for the "creator" to continue designing. Each candidate solution is evaluated according to the relation between the graphical elements in logotype and the relation between the graphical elements and the criteria added by the user at the start. At this stage, the value of the relationship between each element, and between an element and a sensation, was predefined by us and introduced in the system via an external database.

Conclusions and Future Work

Although this project is still a work in progress, and the outcomes need to be evaluated in a real-world scenario, the current outcomes present a good basis for the continuation of work on this topic. Future work will include the increase of element sets in the system and the inclusion of autonomous learning modules in the system.

Acknowledgements

This research is partially funded by Fundação para a Ciência e Tecnologia (FCT), Portugal, under the grant SFRH/ BD/132728/2017.

References

Armstrong, H. and Stojmirovic, Z. 2011. *Participate: Designing with User-Generated Content,* New York, NY: Princeton Architectural Press.

Blauvelt, A., 2011. "Tool (Or, Post-production for the Graphic Design)." In *Graphic Design: Now in Production,* Minneapolis, NN: Walker Art Center. 22-31.

Gerstner, K. 1964. *Designing Programmes,* London: Alec Tiranti Ltd.

Shaughnessy, A., 2012. "When less really does mean less: The post-graphic design era." In *Scratching the Surface: Essays ,*London, Unit Editions. 42-44

Computationally Generating Images for Music Albums

Miguel Cruz, Paul Hardman, João Miguel Cunha
CISUC, Department of Informatics Engineering,
University of Coimbra
migmcruz@gmail.com,{phardman,jmacunha}@dei.uc.pt

Abstract

Nowadays, music albums are seen as a set of unrelated pieces rather than a whole. We propose an approach for a system which produces images for songs of music albums, with the goal of establishing graphic cohesion. It uses semantic analysis of lyrics and semiotic properties to visually represent the meaning and emotions of songs.

Introduction

In the last 20 years there were a lot of changes in the music industry, due to the tremendous technological growth. The new paradigm, dominated by singles and playlists, changed the concept of music album from something which was normally seen as a whole into a simple set of individual and unrelated pieces. Moreover, the relation between the music album and its related concerts has been reduced – more importance is given to festivals with several music artists rather than to individual concerts (Tschmuck 2012).

The main goal of this project is to use computational means to integrate all the materials and events related to a music album, making it work both individually and as a whole. In order to achieve this, we propose a computational approach that produces images from the song lyrics of an album and transforms them according to sound variables of the songs. These images can then be used to (i) create the graphic materials of the album, (ii) produce videos for the songs and (iii) visual effects for the corresponding concerts. This way, all of the materials and events will be integrated in one single graphic system, fully related to the music album.

Our Approach

The relation between text, image and sound serves as a basis to the proposed computational system, which is divided into six layers (L1-L6) . This modular architecture allows an iterative development towards our final goal: having images that illustrate the songs' lyrics and are distorted according to the sound of the voice and instruments, using semiotic properties to visually represent the emotions of the songs.

L1 Lyrics Analysis The first layer is responsible for the analysis of the lyrics, which retrieves words that can be searched in an image database. In order to do this, the system uses the Processing RiTa library to divide the lyrics in nouns, adjectives and verbs.

L2 Lyrics Preparation In this layer, the user is asked to determine the time in the song in which each retrieved word appears, in order to know when the images should be created or distorted. This achieves synchronization between song and image.

L3 Image Retrieval The nouns retrieved by L2 are used in a search process conducted in an image database (e.g. Unsplash), gathering images that illustrate them. Our goal is to follow a semi-automatic approach, in which the user has the possibility of selecting which gathered images should be used.

L4 Image Preparation This layer uses the images provided by the previous layer and applies a filter to them, making them better suited for visual transformations. Currently, we are using OpenCV Canny Edge Detector to transform the images into a set of ellipses. We plan to explore other possibilities in the future: e.g. line representation or even clustering-techniques.

L5 Illustration Production The graphic elements, produced by the previous layer from the gathered images, are used in combination with semantic analysis to produce illustrations. This process uses the verbs and adjectives retrieved using RiTa and uses them to apply visual transformations, based on semantic-semiotic mappings (e.g. words related to movement affect element positioning).

L6 Background Variation The background of the illustration will be affected by sound variables of the instrumental. One example is changing the colour according to the musical scales or chords that are being played at each moment: colder colours if the scales/chords are minor and warmer colours if they are major. In order to do this, we will orientate the study to the piano to make a MIDI analysis in Pure Data. The program could have another component consisting in an estimated analysis of the full instrumental.

Future Work Currently, the user is manually searching the images, as the image retrieval is not yet implemented. In the future we intend to explore with other filters for the Image Preparation layer.

References

Tschmuck, P. 2012. Creativity and innovation in the music industry. In *Creativity and Innovation in the Music Industry*. Springer. 225–251.

Steering Circularity Between Musical Anticipation and Repetition

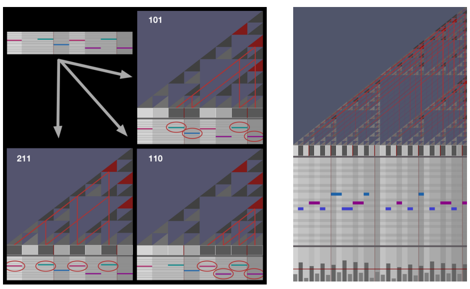
Jay Hardesty
Independent Researcher
Zurich, Switzerland
jayhardesty@gmail.com

Abstract

An approach to computer-aided improvisation that leverages aspects of low-level rhythmic coherence is demonstrated. A connection to number theory provides a self-similar map of rhythmic building blocks, enabling real-time rhythmic analysis and manipulation. The result is real-time navigation and manipulation of rhythmic patterns.

Introduction

This show-and-tell session provides an overview of interrelated systems for computer-aided music composition and improvisation presented at MUME workshops held in conjunction with ICCC in 2016 and 2018. The systems inject musical variations and hybrids of individual musical parts into the context of existing musical compositions. Music generation is steered during playback by human action and judgement.

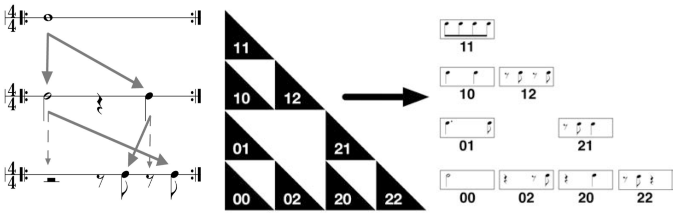


Goals and Systems

The aim is to harness the structure by which listeners make sense of rhythmic patterns, in order to steer composition of new patterns. The assumption is that musician practiced in improvisation navigate these structures intuitively, and that algorithmic co-processing can provide some degree of that capability in the form of a real-time computer interface.

Algorithmic method

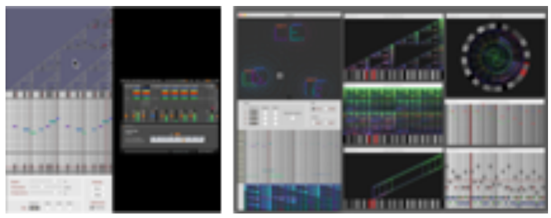
This system detects and manipulates expectation-based configurations composed of nested anticipation outcomes. An emergent map of rhythmic building blocks results from a correspondence between number theory and a tiny set of generative music operations. These building blocks encapsulate nested circularity between anticipation and repetition.



Modes of Interaction

A pair of macOS apps implement these algorithms, extracting and reinjecting note patterns into Ableton Live. One app affords direct manipulation of structure within an individual Live clip, the other app provides a landscape for morphing between multiple Live clips.

The software does not model compositional strategies or evaluate musical results; those activities are left in the hands and ears of the user.



References

Hardesty, J. 2016. Building Blocks of Rhythmic Expectation. Proceedings of the Fourth International Workshop on Musical Metacreation, Paris.

Hardesty, J. 2018. Navigating Outcomes of Rhythmic Anticipation. Proceedings of the Sixth International Workshop on Musical Metacreation, Salamanca.

C

I

ICCC'18

June 25 — 29
Salamanca, Spain

9th International Conference on Computational Creativity

Association for Computational Creativity

International Conference on Computational Creativity