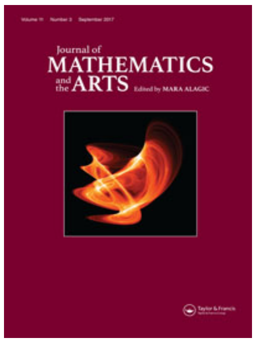


Journal of Mathematics and the Arts

ISSN: 1751-3472 (Print) 1751-3480 (Online) Journal homepage: https://www.tandfonline.com/loi/tmaa20

A taxonomy of generative poetry techniques

Carolyn Lamb, Daniel G. Brown & Charles L.A. Clarke

To cite this article: Carolyn Lamb, Daniel G. Brown & Charles L.A. Clarke (2017) A taxonomy of generative poetry techniques, Journal of Mathematics and the Arts, 11:3, 159-179, DOI: 10.1080/17513472.2017.1373561

To link to this article: https://doi.org/10.1080/17513472.2017.1373561

Published online: 11 Sep 2017.

JOURNAL OF MATHEMATICS AND THE ARTS, 2017
VOL. 11, NO. 3, 159–179
https://doi.org/10.1080/17513472.2017.1373561

A taxonomy of generative poetry techniques

Carolyn Lamb, Daniel G. Brown and Charles L.A. Clarke

David R. Cheriton School of Computer Science, University of Waterloo, Waterloo, Canada

ABSTRACT
We describe computer-generated poetry techniques in the categories of mere generation, human enhancement and computer enhancement. We classify the different kinds of algorithm used to create poetry, and argue that the artificial intelligence techniques used by computer scientists are artistically relevant. We also use computer-generated music as an example to show that this three-level taxonomy applies to computational creativity in other fields.

ARTICLE HISTORY
Received 5 January 2017
Accepted 28 August 2017

KEYWORDS
Poetry; generative poetry; computational creativity; natural language; generative art

AMS SUBJECT CLASSIFICATION:
68-02; 68T50

1. Introduction

Since the middle of the twentieth century, a variety of different people have generated poetry using computers, from artists exploring the effects of algorithms on language, to Internet hobbyists, to computer scientists interested in making artificial intelligence creative. Computer-generated poetry can be an intriguing curiosity, a satire, a meeting of AI and performance art, a serious artistic attempt to explore the possibilities of random and arbitrary text, an attempt to entertain and impress computer users, or an exploration towards concrete advances in computational natural language generation. This multiplicity of purposes leads to a wide variety of different authors in different communities, with different goals and interests, who do not necessarily communicate with communities other than their own. However, after examining many examples of computer-generated poetry, we find that the techniques used to generate such poetry can actually be boiled down into a few simple categories with well-defined relationships.

We define these categories as mere generation, human enhancement and computer enhancement. In mere generation, a computer produces text based on a random or deterministic algorithm. Figure 1 illustrates these categories and some of the major techniques within each one. With the exception of recurrent neural nets, all generative poetry systems we have come across use some form of mere generation as a baseline. However, in the remaining two categories, the results of mere generation are modified and enhanced. This occurs either through interaction with a human (human enhancement), or through the use of optimization techniques and/or knowledge bases by the computer (computer enhancement). The results of mere generation can appear nonsensical, though this is not always a

CONTACT Carolyn Lamb c2lamb@uwaterloo.ca

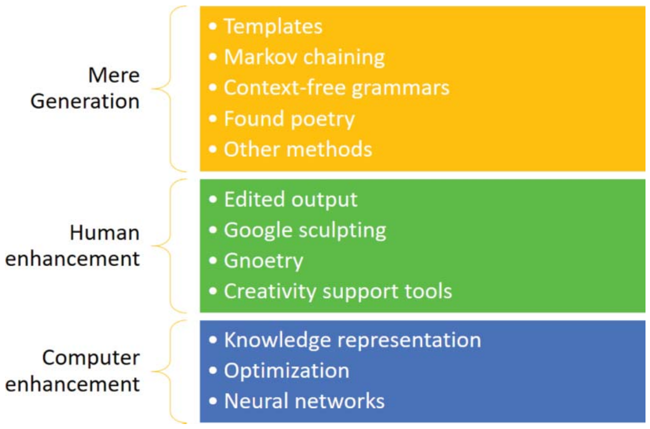


Figure 1. A diagram illustrating our three-part taxonomy.

bad thing from an artistic perspective. By bringing in knowledge about words and the world, and by setting artistic goals, both human and computer enhancement drive generative poetry towards coherence and artistic style.

In the rest of this paper, after discussing prior work and our view of poetry, we illustrate with examples the techniques used in our three categories. We explain why scholars have moved away from mere generation, and argue that computer enhancement, while pursued primarily by scientists rather than artists, has the potential to solve *artistic* dilemmas in generative poetry. We then briefly bring up the related field of generative music, to show that our taxonomy can be generalized to other creative tasks besides the generation of poetry.

1.1. The goals of poetry

As computer scientists, we should clarify what we mean when we talk about artistic perspective, artistic style or artistic goals. Human poets write according to a variety of complex goals, the full spectrum of which are beyond the scope of this paper. According to the Princeton Encyclopedia of Poetry and Poetics, a few of the most common poetic goals include:

- Describing or imitating reality (the Mimetic Theory)
- Having a specific effect, such as educating or inducing an emotion, on readers (the Pragmatic Theory)
- Communicating the poet's emotions (the Expressive Theory)
- Creating a work of art that exists for its own sake (the Objective Theory)

[1]

These goals are very broad and each can be conceptualized and implemented in a variety of ways. Most ways of achieving most of the goals will require, as a necessary but not sufficient aspect, that a poem is intelligible to an educated human reader. By understanding what is in the poem, the reader can more easily recognize the description of reality, understand the emotions of the author, experience the poem's intended effect or appreciate the poem's inherent value. For the purposes of this paper, when we speak of mainstream artistic goals, we are referring to this kind of poem in which the reader is meant to understand something meaningful.

It is, of course, possible to write a poem which is valuable without being understood. Indeed, this is the explicit goal of several twentieth century poetic movements, including the Dadaist movement, which uses intentionally nonsensical text in order to express rebellion against language's rules and traditions [3]. Intuitively, computer-generated poetry might seem to be more compatible with these rebellious movements than with mainstream poetry. There is nothing wrong with rebellious poetic movements, but part of our argument in this paper is that computers using AI techniques have the potential to generate content that is meaningful to humans and thus enter the mainstream. We will be returning to this mainstream/rebellious distinction throughout this paper.

Digital poetics encompasses a wider range of techniques than those described here. Computer systems allow a variety of new techniques to human poets including hypertext poetry, kinetic poetry, chatbots as art, interactive fiction [12], multimedia poetry and even poetry that presents itself as a game [16]. However, for the purposes of this paper, we are interested only in poems representable by a text file, in which the computer has a meaningful role in determining what the text will be. This is our working definition of 'generative poetry'. We will be focusing primarily, though not exclusively, on generative poetry in the English language.

1.2. Prior work

We are not the first to attempt a taxonomy of generative poetry. Roque [54] classifies poems according to the goals of their creators, while Funkhouser [15] uses the categories of permutational, combinatorial and template-based generation. Gervas [18] divides what we would describe as the computer enhancement category into four subcategories based on the type of artificial intelligence techniques used. These taxonomies are useful. However, our taxonomy serves needs that others do not. It includes and takes seriously generative poetry from a variety of sources, whether scientific, hobbyist or artistic, and focuses not on technical descriptions of processes but on the purposes for which these processes are used.

The closest existing taxonomy to ours comes from Ventura's very recent paper [64], where he classifies computationally creative systems by the sophistication of the computational techniques used. Ventura's taxonomy is abstract and broad, and aimed at describing the degree to which a system is creative. Ours is focused specifically on poetry, including forms of poetry not made with computational creativity in mind. However, Ventura's taxonomy is a good counterpart to ours, and we will refer to it throughout this paper.

There are two reasons why our taxonomy is useful. First, for computer science researchers who are interested in poetry, this paper provides a brief survey of what has been done before in terms of methods and goals. A researcher who wants to show that their poetry generation system performs non-trivial operations can start by understanding the difference between mere generation methods and enhanced methods. Second, for readers interested in the artistic and humanistic side of creativity, our taxonomy shows what effects are possible using computational methods. Furthermore, we use our taxonomy to argue that enhanced AI poetry techniques should be considered relevant to poets

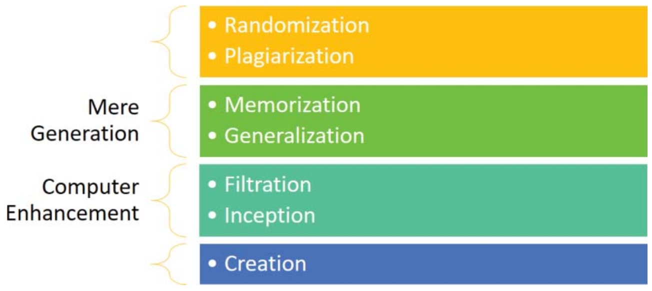


Figure 2. A diagram illustrating Ventura's taxonomy and its relation to ours. Outer labels ('mere generation' and 'computer enhancement') are ours, not Ventura's, and are meant to illustrate areas of overlap between the two taxonomies, not to imply that Ventura would necessarily use the terms in this manner. Note that Ventura's taxonomy, unlike ours, explicitly proceeds in a hierarchy from the least creative (top) to the most creative (bottom) methods. It includes some categories which we have not observed in actual generative poetry systems, but does not include anything corresponding to human enhancement.

as well as researchers. Figure 2 illustrates the relationship between Ventura's taxonomy and ours.

2. Mere generation

As Ventura points out, 'mere generation' is a term with a long and loaded history in computational creativity [64]. Many researchers claim that they want to transcend mere generation, but the term is often used without a precise definition. For the purposes of this paper, when we refer to mere generation, we refer to systems in which *the computational element* of the system is either random or arbitrary. Humans may have painstakingly handcrafted the system to increase the likelihood that the random calculation results in a pleasing output, and the randomness may be taken from a non-uniform distribution; but the *computational process* used by the system is random or arbitrary. This implies two issues: first, from a computer science perspective, the systems are not computationally interesting. Second, from a cognitive science perspective, the systems are not interesting because they do not model the cognitive steps theoretically associated with creativity. However, from an *artistic* perspective, there is no a-priori reason to assume that the art produced by a mere generation system is inferior to the art produced in another category. If we wish to make such an artistic claim, we will have to look at the poems empirically.

Our definition of mere generation corresponds to the first few categories in Ventura's model: Randomization (in which elements of the creative product are chosen completely at random), Plagiarization (in which existing examples of the creative genre under consideration are copied), Memorization and Generalization. In Memorization and Generalization, the system attempts to capture and replicate certain properties of a training set of human-made work, without duplicating the human-made work entirely, but no knowledge-based or goal-directed reasoning is performed. Note that in our research we have not come across any poetry systems simplistic enough to correspond to the Randomization or Plagiarization categories.

By using the term ‘mere generation’, we do not mean to imply that there is necessarily anything ‘mere’ or trivial about either the poems or the human artistic process that goes into making them. We use the term not to belittle but to distinguish these systems from more algorithmically complex ones.

In the rest of this section we will list methods used by mere generation systems. Note that not every system using one of these methods is a mere generation system. Systems that *straightforwardly* apply the methods below, without further processing and enhancement, are mere generation systems. This includes all the specific systems that we use below as examples. As we will describe in the following sections, most human enhancement and computer enhancement systems begin with a mere generation method and build on top of it.

2.1. Methods of mere generation

2.1.1. Templates

Template-based generation, also called slot-filling, has been common since the first generative poetry program, Theo Lutz’s ‘Stochastic Texts’ [37]. Template generation is one of the simplest means of constructing a poem. The basic steps are as follows:

(1) Create lists of words or phrases in different categories, e.g. nouns or verbs (though the categories need not be based on a part of speech).
(2) Create one or more line templates with slots into which a word from a given list can be inserted.
(3) Randomly select a word from the appropriate list to fill each slot.

This process corresponds to Ventura’s category of Generalization. Essentially, the template models poetry as a sequence of types of words in a particular order, and randomly creates new work to fit that model.

‘Stochastic Texts’ uses templates and word lists based on lines from Kafka’s ‘The Castle’, resulting in lines like the following [37]:

NOT EVERY LOOK IS NEAR. NO VILLAGE IS LATE.
A CASTLE IS FREE AND EVERY FARMER IS FAR.
EVERY STRANGER IS FAR. A DAY IS LATE.
EVERY HOUSE IS DARK. AN EYE IS DEEP.
NOT EVERY CASTLE IS OLD. EVERY DAY IS OLD.

Template poetry can draw its word lists from existing art in this way, or from dictionaries representing the whole of the language, or the lists can be handcrafted by the programmer. Slots in a template need not be filled by a single word; the word lists can just as easily contain phrases or other structures.

A major issue in template poetry is repetitiveness. Often, running a template program several times will produce a repetitive effect in which the template’s structure becomes obvious, as in this example (from ‘The House’, by Alison Knowles and James Tenney [34]):

A HOUSE OF STEEL
    IN A COLD, WINDY CLIMATE

```
        USING ELECTRICITY
            INHABITED BY NEGROES WEARING ALL COLORS
A HOUSE OF SAND
    IN SOUTHERN FRANCE
        USING ELECTRICITY
            INHABITED BY VEGETARIANS
A HOUSE OF PLASTIC
    IN A PLACE WITH BOTH HEAVY RAIN AND BRIGHT SUN
        USING CANDLES
            INHABITED BY COLLECTORS OF ALL TYPES
```

Such repetitiveness may, as in the given example, be intentional. However, most poets want output that looks fresh each time the program is run. One way of achieving this is with templates that change over time. John Morris, for example, uses shifting templates to create haiku without obvious syntactic repetition [45]:

Frogling, listen, waters
Insatiable, listen,
The still, scarecrow dusk.

Listen: I dreamed, was slain.
Up, battles! Echo these dusk
Battles! Glittering ...

2.1.2. Markov chaining

A Markov chain is a statistical model applied to data in a sequence. Based on the last N entries, an Nth-order Markov chain calculates the probability distribution for the next entry. The series of N entries used to make the prediction are referred to as an *n-gram*; a critical feature of Markov chaining, which keeps computational costs low, is that no entries before the n-gram need to be considered. Based on the entries in the n-gram, the Markov chain calculates the probability of different entries appearing next. It then probabilistically generates the next entry, which produces a new n-gram (including the newly generated entry, but not the first entry in the old n-gram). Further entries can be generated in the same way indefinitely.

For poetry generation, the entries in the n-gram can be characters or words. A probability model is generated either from a broad corpus or a specific work. Based on a few user-generated starting entries, the system repeatedly samples from the model to create the next entry. Using words as entries ensures that no novel or partial words appear in the output. Using characters results in many non-words and neologisms, which can be an intended effect, as in this example from Dissociated Press by R.W. Gosper [54]:

```
book her sist be chin up seen a good deal uneasilent for coursation
dropped, and the
litter on,
    The Queen was
siliarly with them, the Footmance.
    Would guess, an't
grom one foot to thistle, to keep and reachinah'll be she could not ever who had not atte-
book
hastily. Convers began to trings; into thing on wast the door, and
the.
```

Markov chaining is also an example of Ventura’s Generalization stage. In this case, instead of a specific ordering of types of words, the system’s model of poetry is the set of probability distributions forming the Markov chain, which is generated by copying the probability distributions of an existing text.

Markov chain poetry is related to Dadaist ‘cut-ups’, invented by Brion Gysin, in which a text is cut into N-character blocks and rearranged [15]. While Markov chains preserve many features of the input text, they fail to replicate grammar. Even with word-based models, output tends to meander incoherently from one topic to another. This wandering feature can be used to artistic effect. For example, a corpus built from two contrasting sources can produce language that switches strikingly from one style to another, as in the case of Charles Stross’s Lovecraft.pl [58], which merges a Markov chain derived from the King James Bible with one derived from the horror fiction of H.P. Lovecraft.

2.1.3. Context-free grammars

Poets who want something more grammatically correct sometimes turn to context-free grammars, a class of models from computational linguistics. A context-free grammar constructs sentences by recursively applying generation rules. It consists of terminal symbols (the actual words or phrases that occur in a finished poem) and non-terminal symbols. Each non-terminal symbol corresponds to a set of terminal and/or non-terminal symbols with which it can be replaced. A non-terminal symbol can also be replaced by a group of symbols. During generation, the system begins with a single non-terminal symbol. It then runs replacement operations repeatedly. At each step, every non-terminal symbol is replaced with an entry in its corresponding set of symbols. The process concludes when no non-terminal symbols remain.

Compared to Markov models, context-free grammars are a better representation of the recursive structure of human language. For example, the non-terminal symbol [NOUN PHRASE] can be replaced with a noun, or with other structures that take the grammatical place of a noun. These structures can become arbitrarily complex (‘the boat that the man in the green suit rowed across the river yesterday’) and still be grammatically correct in the subject portion of an English sentence.

By adding Markov chain-like probabilistic reasoning to a context-free grammar, one can construct a stochastic context-free grammar which constructs the most likely sentences based on some input corpus while remaining syntactically correct. Jim Carpenter’s Electronic Text Composition uses probabilistic grammar, resulting in language which is semantically odd, yet more coherent than the output of a Markov chain [8]:

The important statement, like one advance act.
A spit goes eastern, showering.
it perches on the branching foam
The statement.

2.1.4. Found poetry

Another mere generation method is to skip the generation process and, instead, use a computer to harvest text written by humans. While most generation techniques make some use of pre-existing human language – as a corpus for calculation of N-gram frequencies, for example – found poetry preserves entire human-written sentences without significant modification. The computer's role is to select text which meets some constraint and present it, perhaps juxtaposed with other texts, outside its original context. Examples include Ranjit Bhatnagar's 'Pentametron', which constructs sonnets by assembling pairs of rhyming, 10-syllable posts on Twitter [6], and the New York Times Haiku project, which mines phrases with a 5-7-5 syllabic structure from that newspaper [27]:

> Surely that shower
> couldn't have been going since
> yesterday morning.

It is not clear exactly where found poetry would fit in Ventura's classification. Arguably, since the found poetry systems described here contain encoded knowledge about poetry (for instance, the number of syllables required), they too would belong in the Generalization category. However, if a found poetry system used actual poetry of the appropriate type as source material, it would quickly devolve into simply reproducing this poetry (Plagiarism from Ventura's classification) or, perhaps, reproducing it mostly with some errors (which would be the Memorization level).

2.1.5. Miscellaneous methods

Some poetry is generated using other mere generation methods. Poetry can be constructed by, for example, selecting words which contain certain combinations of letters and arranging them on the screen [16]. Or the words in a short phrase can be permuted, as in Brion Gysin's 'I Am That I Am' [15]. Edde addad's poem 'Disappointment and Self-Delusion' is constructed around mistakes made by the speech recognition software Dragon NaturallySpeaking [54]. We will not explore these methods in detail here. Since the linguistic processes done by the computer are random or arbitrary, they are still classed as mere generation.

2.2. The problem with mere generation

While the methods described above may seem endlessly flexible, there is a limit to what they can do. The frequency of words and the rules of grammar can be modelled, but semantic coherence is more difficult: the poems are valid sentences or language fragments, but do not really mean anything individually or as a whole. This is an obstacle to any poet who wants their poetry to meet a mainstream poetic goal and be understood by readers.

For this reason, some pioneers and experts in generative poetry have grown discouraged with the form. Charles Hartman, for example, after years of creating generative poetry, concluded that even the most syntactically correct programs 'did only a little to drive the random words toward sense' [15]. While poetry generated using templates can make sense, it is difficult for a template to retain interest and meaning after repeated use exposes its pre-set structure. Chris Funkhouser wrote in 2012 [16]:

> Researching the topic for nearly twenty years, I have encountered dozens of poetry generators. Many have issued convincing poetry, but even the best of them fatigue the reader with blatant slotted structures and repetition.

Lack of coherence is not necessarily a problem for everyone. As mentioned, Dadaists and other poets are interested in generating nonsensical language [3,15]. Similarly, Oulipians and conceptual poets [23] are interested in inventing new poetry techniques, not in the content or quality of the poetry itself. This is only to mention a few of the rebellious schools of poetry for which a poem failing to make sense is not a problem. However, poetry that becomes mechanically repetitive will eventually fail to shock anyone. For this reason, many new media poets have turned their focus away from the generation of poetic text as such, and towards other uses of computers, such as multimedia poetry [16]. Others look for ways to improve on mere generation. One such method is for a human to edit the output; the other, more ambitious but still in its infancy, is to improve generation through artificial intelligence.

3. Human enhancement

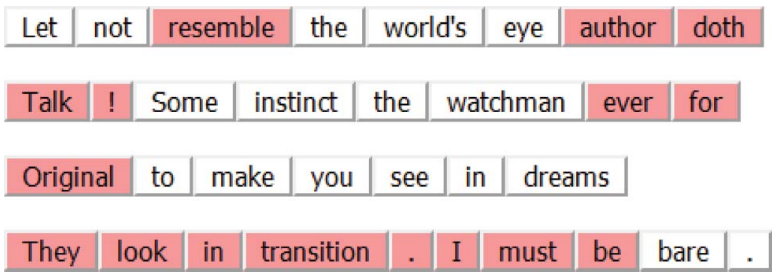
The most obvious way for a human to enhance computer-generated poetry is to edit the poetry generator's output. While this arguably invalidates the generator's usefulness [9], it is an established practice. John Cage, for instance, removed unwanted words from the output of his algorithms [15]. Computational text generation is seen by many as a 'jumping-off point' [9] from which they acquire raw material. While almost any poetry generation system involves a human programmer selecting the best examples of output for publication, we define the human enhancement category as poetry in which post-algorithmic human involvement goes beyond selection and into actually modifying the output text. (Human–computer co-creativity is not mentioned in Ventura's taxonomy, as Ventura is concerned only with judging the sophistication of the computer.)

Human enhancement is a common technique for poetic groups outside the mainstream. One such group is the Flarf movement. Flarf revolves around intentionally inappropriate, lowbrow language harvested from the Internet. One Flarf technique, 'Google sculpting', consists of searching the Internet for particular terms, taking phrases from the results (a form of found poetry) – and then recombining and modifying these phrases however the poet sees fit. The result is a distinctive, over-the-top poetics – as in this example, which was published in the prestigious magazine Poetry [26]:

Oddly enough, there is a
'Unicorn Pleasure Ring' in existence.
Research reveals that Hitler lifted
the infamous swastika from a unicorn
emerging from a colorful rainbow.

Even more interesting is Gnoetry, an application for interactive text generation which allows decisions to cycle between the computer and a human user. The computer generates poetry based on n-grams from a user-provided corpus. The user can click on individual generated words, deciding which words and phrases are worth keeping and which should be generated again. The computer then generates new phrases to replace those the user did not find satisfactory. This repeats as many times as the user would like. An example of what Gnoetry looks like during the generation process is shown in Figure 3.

Gnoetry turns generation into a dialogue between the human and computer. It is a dialogue that plays to the strengths of both participants: the computer can endlessly and


Let not resemble the world's eye author doth

Talk ! Some instinct the watchman ever for

Original to make you see in dreams

They look in transition . I must be bare .

Figure 3. A screenshot of Gnoetry in action. Words in white have been selected by the human user as words to keep, while pink words will be replaced at the next generation step.

tirelessly recombine words, while the human can use their superior judgment to select the most promising combinations. An online community, blog and several chapbooks exist showcasing the work of various poets using Gnoetry. A favoured technique, as with Markov chain poetry, is to mix together the styles of several contrasting works [14].

The notion of human and computer cooperation also appears in the development of creativity support tools. Kantosalo et al., for example, have a system to help elementary school students write poetry, in which the computer suggests possible words in a magnetic poetry format [31]. The goal here, rather than showcasing the poetic skill of a computer, is to scaffold human learning of poetic skills by having a computer assist in some of the poetic decisions. A program like Gnoetry can also be used to test a human's poetic skill by requiring them to work within the computer's constraints.

4. Computer enhancement

The other way to enhance computational poetry is to add more advanced concepts from computer science: not only generating words, but making sophisticated attempts to optimize the output. This set of methods comes not from the humanities but from scientists in the discipline of computational creativity. While a variety of AI techniques can be brought to bear on these problems, there are two main purposes to which these techniques are put. One is optimization of the system's poetic output on some metric; the other is connection of the generation apparatus to underlying knowledge about the world.

Nearly all computational poetry systems we analysed begin with one of the mere generation techniques discussed above. Template generation is most common [10,46,59–61], but the McGonagall system [39–41] uses a technique similar to context-free grammar. Barbieri et al.'s system uses an improvement on Markov chains [4], while DopeLearning [38] (recombining lines from rap songs) and our own TwitSong [35] (recombining lines from Twitter) are found poetry systems. What distinguishes all these systems from mere generation systems is that something – optimization, a knowledge base or both – is *added* to the basic technique. This addition can be incorporated into the mere generation step as guidance, or can be added after that step is complete.

Ventura's classification refers to the use of optimization metrics as Filtration, and the use of a knowledge base as Inception. Ventura considers Inception to be a higher level of creative sophistication than Filtration [64]. Ventura also provides an additional category, Creation, in which a computer's work is informed by direct perceptual processing. To the best of our knowledge, no poetry system has yet been constructed in this way.

4.1. Data mining and knowledge representation

Merely generated poetry tends towards nonsense; whether artistically desired or not, this is a result of the computer's lack of real-world knowledge. By representing semantic facts, an artificially intelligent system can attempt to overcome this limitation. Just as Gnoetry puts human enhancement inside the generation process rather than adding it on after generation, a knowledge base can be put inside the generation process to guide its range of output. Creative humans, too, require knowledge about the world and their domain as groundwork for apparently spontaneous creative insights [5,33,53,55].

One way of representing commonsense knowledge is to encode it in propositional logic. One experiment in this vein is Ruli Manurung's McGonagall system [40]. McGonagall uses the AI technique of chart generation which, given an input meaning, lexicon and set of grammar rules, can exhaustively generate all grammatically correct representations of the input meaning. McGonagall generates poetry in a metrical form by encoding metrical constraints as grammar rules. When the input meaning is directly encoded by humans, this process results in metrically correct and very logically consistent poetry [40]:

The cat is the cat which is dead.
The bread which is gone is the bread.
The cat which consumed
the bread is the cat
which gobbled the bread which is gone.

However, propositional logic alone does not solve the sense/nonsense problem. McGonigall's poetry only makes sense when the exact intended meaning of the poem is directly encoded by a human. When the system is allowed to construct its own propositions, it immediately lapses back into nonsense [41]:

They play. An expense is a waist.
A lion, he dwells in a dish.
He dwells in a skin.
A sensitive child,
he dwells in a child with a fish.

It is not enough for a computer to be able to represent facts; to be anything other than nonsense, these facts must have some relation to human experience. Real-world propositions could be added to a system like McGonagall through semantic parsing of a corpus of human-written texts. A few systems in non-English languages have made good strides expanding their bank of propositions by taking advantage of existing semantic knowledge bases such as ConceptNet [52,57] and WordNet [2]. But to the best of our knowledge, English-language systems have not yet followed suit.

What *has* been done in English is parsing of word associations. By calculating the co-occurrence of different words in a source text, computers can gain a sense of which words are and are not related to a given topic. This can guide the computer towards the construction of lines and sentences that relate to each other in sensible ways. It is important to note that word associations in this sense are different from the N-grams used to create a Markov chain. Markov chaining only measures words that appear immediately after each other. Word association does not take into account word order, but instead looks at what meaningful words are used in a given chapter, article or paragraph which are different from the words in other units of the text. In this way, word association mining gives a sense of what words are topically associated with each other or have similar meanings.

Toivanen et al. create poetry using word associations mined from Finnish Wikipedia [59] and news stories [60]. Netzer et al. [46] use a list of word associations from psychological testing, which they claim produces more intuitive, human-like semantic connections than word associations from an encyclopedia. Combining these associations with syntactic templates results in plausible haiku [46]:

cherry tree
poisonous flowers lie
blooming

Veale and Yao [63] search Google N-grams for similes, which (as well as being a poetic device in their own right) contain implicit information about the properties of things in the real world. Their system then uses this implicit information to create new similes of its own. The Full-FACE system [10] modifies these similes to create full poems. Its poetry, while repetitive, is full of comparisons that make sense to a human [10]:

the wild relentless attack of a snake
a relentless attack, like a glacier
the high-level function of eye sockets
a relentless attack, like a machine
the low-level role of eye sockets
a relentless attack, like the tick of a machine

4.1.1. Optimization

The other mode of enhancement that computer science has to offer is optimization. Given some formal definition of the desired traits of a poem, a computer system can begin to, in some sense, think critically – testing different possibilities and choosing the ones which best fit its requirements. While this is a very elementary form of critical thought, it is an important step towards true creativity on the part of computers: being able to understand one's own aesthetic and create work to match. Creative humans engage in their own forms of optimization, continually re-evaluating their work during the creative process and comparing it to their goals (*cf.* [11,22,67]).

Optimization techniques applied to generative poetry include stochastic hill-climbing search [39], generate-and-test [10,46], genetic algorithms [19,41], constraint satisfaction [61], case-based reasoning [17], dynamic programming [69], statistical machine translation [28] and recurrent neural networks [38], the last of which we will look at in more detail in a later section. The details of these methods are beyond the scope of this paper. In brief, each one starts with a mere generation technique and a goal. The system evaluates how close the generated text is to meeting its goal, and then re-generates or modifies its text repeatedly until it meets the goal as closely as possible. Any mere generation technique can be combined with most optimization techniques. A few optimizations are designed for a specific mere generation technique, such as elementary Markov constraints

[48] and constrained Markov processes [4], both of which allow a programmer to impose structural requirements onto a Markov model.

What goals do these poetry systems work towards? Many concentrate on basics such as meter, rhyme and grammaticality [39,46,61] or fitting generated text to a rhythm [24]. However, a more exciting possibility is setting goals for the subject matter, emotions or artistic style of a poem – traits which can be measured through natural language processing techniques. It is important to note that if a computer is judging the subject matter of its generated text, then optimization has been combined with knowledge representation: without some form of implicit or explicit knowledge representation, it would be impossible to process a text's semantic traits. A great deal of Computer Enhanced poetry uses both techniques, and thus would be considered Inception under Ventura's model.

As an example, ASPERA [17] includes mood as one of its constraints. DopeLearning [38] generates rap lyrics according to a variety of textual measures, including the maximization of internal and multisyllabic rhymes, and uses a neural net to represent semantic content. The Full-FACE system [10] chooses between possible goals, including relevance to the poem's topic, emotion and some stylistic measures such as 'flamboyance' (the number of unusual words). Yan et al. [69] maximize the relevance, importance and coherence of individual words in each poem, while He et al. [28] optimize coherence based on mutual information. TwitSong [35] also selects tweets according to their topic relevance, sentiment and the presence of sensory imagery: this results in poems composed of tweets which, while written by different people at different times, nonetheless come together into something reasonably coherent [35], and presumably more effective than the random choices of Pentametron [6]. An example, TwitSong stanza is:

Hey Nashville...2014 is pretty awesome!
Happy 2014 friends! Be safe out there!!
Had a great New Years Eve at Magic Kingdom
We started off 2014 with a prayer

A particularly sophisticated optimization technique is that of Gervás's WASP system, which creates Spanish-language poetry through an evolutionary approach involving many modules which create, evaluate and edit candidate poetic texts [19,20]. 'Babblers' are the system's mere generation layer, which create baseline text using an N-gram model. 'Reviser' modules can make specific changes to the text aimed at correcting the problems detected by a 'judge'. This is the closest to a human poet's revision process that we have encountered in the field of generative poetry. The WASP system's optimization goals include poem length, verse length, rhyme, stress pattern, similarity to source (poems that plagiarize the source material are rejected) and fitness for a specific poetic form. Some of WASP's poems are successful enough to have been published in a book about generative poetry [20].

4.2. Neural networks

A final computer enhancement method, recurrent neural networks, is complex enough to deserve a discussion of its own. Neural networks are a machine learning technique in which the computer, by mimicking the activities of human neurons, learns its own high-level encoding of input data. Like a verbal equivalent of Google's Deep Dream [44], neural networks trained on poetry can generate a high level, general encoding of the patterns in the poetry they have seen and then reverse their processing to generate new work of their own. A key property of neural networks, however, is that their encoding is implicit. Information is contained in the strengths of thousands of connections between neuron-like nodes, rather than existing in a form which can be easily communicated to humans. Because neural networks are 'black boxes' in this sense, it takes some thought to discern where in our taxonomy a neural network-based poetry system would belong.

Neural networks have been used to generate poetry in several languages, but are most prominent in the generation of Chinese classical poetry. The most common form of neural network in Chinese poetry generation is a recurrent neural net or RNN, in which the connections between neurons can form cycles [21,72]. To this framework some researchers add features such as long short-term memory [65] and an attention model [65].

To constitute computer enhancement in our taxonomy, a neural network would need to either contain implicit knowledge representation or to be optimized on some metric. Most poetry neural networks are optimized during their training phase: the weights of connections in the network are repeatedly adjusted in order to produce output that most closely meets some metric chosen by the programmer. The most common such metric is cross-entropy, which is a statistical measure of the similarity of character distributions between the output and a source corpus [65,68,72]. The decoding portion of the network, which translates from connection patterns back into human language, can contain additional constraints for goals like rhyme, topicality and tone pattern [21,72].

The argument regarding knowledge representation is slightly less obvious, but we would argue that if a neural network contains global information about word associations, then it contains knowledge representation, just as poetry systems do which acquire this information in another way. In practice, many systems use pre-existing word association corpora to process user-defined keywords before feeding them as input into the neural network [21,66,72].

However, it is possible to imagine a naive application of a neural network which would be mere generation. Imagine a neural network which is trained to produce the next character of a sequence based on the last N characters. It uses an n-gram representation for input which is similar to that of a Markov model, without any global information, and instead of having its connection weights adjusted to optimize its output, it is allowed to discover the patterns in these characters on its own. Such a neural network would contain neither optimization nor knowledge representation. It would be essentially just a very complicated Markov model, and one would expect its output to be similar to that of a Markov model. Indeed, simplistic applications of recurrent neural networks to poetry in English can strongly resemble the output of a Markov model [32]:

> O, if you were a feeble sight, the courtesy of your law,
> Your sight and several breath, will wear the gods
> With his heads, and my hands are wonder'd at the deeds,
> So drop upon your lordship's head, and your opinion
> Shall be against your honour.

The Chinese systems which do incorporate computer enhancement produce strikingly convincing output. At least one Chinese system [66] produced results that non-expert human evaluators could not distinguish from classical Chinese poems. Another [65] produced poems that human experts rated as more ‘poetic’ (meeting the constraints of the chosen poetic form) than human poetry, although it did not meet human standards for fluency or meaningfulness. An expert in Chinese poetry whom we queried agreed that the best published Chinese poems looked similar to something a human could write [36].

The neural network aspect of these systems is often combined with other mechanisms to ensure that they meet all their constraints. In addition to the word association tools already mentioned, Zhang and Lapata’s system [72] adds statistical machine translation to improve the poems’ coherence. Also notable is the use of something resembling planning mechanisms in many systems. Several systems generate an outline of their poem, either using keywords [66], separate layers [72] or a separate neural network [68], before feeding this outline into the central RNN to generate actual lines.

Few systems in languages other than Chinese have replicated these results. An exception is the recent work of Goodwin, who trained an LSTM-RNN (recurrent neural network with long short-term memory) on a corpus of contemporary English poetry [25]:

And still I saw the Brooklyn stairs
with the shit, the ground, the golden haze
Of the frozen woods where the boat stood.
When I thought of shame and silence,
I was a broken skull;
I was the world which I called it...

Goodwin reports that his results with the LSTM-RNN were uneven, and overall semantic coherence is still an issue; the poems are still arguably not ‘about’ anything. But in our opinion, the stylistic rendering is good enough to give the impression of images and moods that a human reader can understand, and of a writer with an ear for rhythm and phrasing at a level no other English poetry system we are aware of has approached. Goodwin’s work has also received a favourable reception from human poets not involved in generative poetry [25].

4.3. *Combined computer and human enhancement*

It is possible to combine both computer and human enhancement. In a combined enhancement system, both the workings of an AI system and the judgment of a human are involved. One example of this type of system is Barbieri et al.’s [4], which generates lyrics using a constrained Markov process, but allows a human to perform the final step in the generation by choosing each verse from a set of five computer-generated candidates.

The science of computational creativity is in its infancy, and these are small steps compared to the complex goal-setting process of skilled human poets. Nonetheless, further refinements in goal specification and knowledge representation could produce truly interesting generative poetry.

5. Separation of generative poetry communities

Researchers working on Computer Enhanced poetry, and artists working on Human Enhanced and Merely Generative poetry, often have little communication with each other.

Computer Enhanced poetry, with the exception of recent work on recurrent neural networks, is typically created by members of the computational creativity research community. These researchers are interested specifically in the computational implementation of creative decisions, and thus, Merely Generative poetry is of little interest to them – in fact, for some researchers, mere generation is anathema to their stated goals [64]. Previous work by such researchers categorizing generative poetry, such as Gervas' taxonomy [18], make no mention of work outside the computer enhancement category at all. While this is logical for researchers with a technical focus, a lack of awareness of generative poetry created by poets and artists outside their field prevents computer science researchers from making compelling arguments about what their work contributes to the field artistically.

Some cultural critics and researchers in the humanities do a better job at presenting a broad perspective. Roque's work [54] places computational creativity projects in context with similar projects from outside that field. However, other important and detailed reviews, such as Funkhouser's [16], make no mention of projects created by computer scientists.

There are many ways to gauge a computational creativity project's success, but producing credibly artistically successful work is one of the goals of most such projects. Cultural critics are ideally placed to provide feedback and critical analysis of computationally creative work on these grounds, provided there is communication between them and computer scientists and awareness of each other's existence. (Success with cultural critics is important from the press perspective of judging creativity – cf. Jordanous's [30] explanation of the four perspectives of creativity evaluation. The press perspective is concerned with a creative work's social success. Cultural critics also have informed comments to make from the product perspective, in which a creative work is judged by its properties after completion.)

Meanwhile, as explored above, computational creativity researchers potentially have something to give back to the humanities – by widening and deepening the scope of what generative poetry systems can achieve.

One very recently emerging community which does bring both sides of the generative poetry community together is Google's Artists and Machine Intelligence program, including an informal conference which was initiated in 2016 [42]. This program is explicitly designed to bring together computer scientists, artists, neuroscientists and psychologists to work together on generative art, including poetry. Institutions such as the University of Helsinki also have ongoing poetic collaborations between computer scientists and working artists [62]. With such multidisciplinary efforts becoming more mainstream and widespread, there is a great deal of reason to hope that researchers on both sides of the art/science divide will continue to come together to make ever more interesting generative poems.

6. Generalization and comparison with music

Our taxonomy is more interesting if its principles can be generalized to other domains. An obvious example would be other domains of digital poetry, such as those described by Douglass [12] and Funkhouser [16]. Certainly many such poems combine the kind of generation that is of interest to us with hypermedia techniques, which can themselves be seen as a form of either human or computer enhancement, applied to the poem's mode of presentation rather than to its words. Nick Montfort and Stephanie Strickland's 'Sea and Spar Between', for example, uses a handcrafted grammar (a mere generation technique) to combine phrases from Emily Dickinson and Herman Melville's work, and displays these phrases using a combinatorial framework which could only be accomplished using the calculating power of a computer [43].

However, in our specific research program we are more concerned with whether our taxonomy is generalizable to computational creativity as a whole. To investigate this question, we will look very briefly at the field of generative music. Our taxonomy – mere generation, human enhancement and computer enhancement – does apply to music, with only small modifications.

6.0.1. Mere generation

Like poetry, music can be composed using a Markov model [13,51]. In fact, Markov models are thought to represent musical style particularly well, at least over short sequences [48]. Other ways of generating music include cellular automata [71], a random trajectory in a directed graph [56], or even producing music from a visual image as if it were a spectrogram [29]. What these techniques have in common is that, as with mere generation in poetry, the creative choices made by the computer are either random or arbitrary. The program simply produces notes according to its rules.

6.0.2. Human enhancement

The human enhancement category in music is most noticeable in jazz improvisation. Computer improvisation systems are built to play alongside human improvisers. They need to process human musical input in real time and respond to that input with novel sequences of notes [7] or with an appropriate synthesis of previously recorded notes and chords [49]. A more avant-garde alternative is the creation of digital instruments. An instrument's responses to human input can be non-obvious or shifting, which results in unpredictable interactions between the musician and the instrument [70]. Like Gnoetry, improvisation systems and digital instruments can create new works through interaction which neither the computer nor the human could have created on their own.

6.0.3. Computer enhancement

As with poetry, the output of mere generation can be fit to hard and soft optimization constraints using techniques such as answer set programming [47], genetic algorithms [56] or elementary Markov constraints, which were in fact specifically designed with music generation in mind [48,50]. Such constraints can be rooted in music theory [47,56], or a specific goal, such as making cover songs incorporate features of the original [51] or imposing a melodic contour chosen by the user [48]. Due to the non-representational nature of music, it is more difficult to find generative music that incorporates a semantic knowledge base. We also see music systems in which human and computer enhancement are combined. Systems can, for example, improvise with human partners and use machine learning to optimize that improvisation [7].

7. Conclusion

In our taxonomy, there are three areas of work in generative poetry: a mere generation technique, human extensions to the technique and computer extensions to the technique. We believe that when critics such as Funkhouser declare that generative art has reached a plateau [16], it is because they are looking only at mere generation and not at the more powerful computational techniques from outside the field which can enhance it. Artistic optimization and knowledge representation techniques have not yet reached their full potential, but they have the power to push generative poetry forward towards the kinds of sense and style that are currently lacking. Far from being a played-out form, generative poetry is just getting started.

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

Canadian Network for Research and Innovation in Machining Technology,Natural Sciences and Engineering Research Council of Canada, grant number RGPIN-2016-03661.

ORCID

Carolyn Lamb http://orcid.org/0000-0003-4384-7589

References

[1] M. Abrams, *Theories of Poetry*, in *The Princeton Encyclopedia of Poetry and Poetics*, R. Greene, S. Cushman, C. Cavanagh, J. Ramazani, and P. Rouzer, eds., Princeton University Press, Princeton, 1974, pp. 639–649.

[2] M. Agirrezabal, B. Arrieta, A. Astigarraga, and M. Hulden, *POS-tag based poetry generation with WordNet*, in *Proceedings of the 14th European Workshop on Natural Language Generation*, Association for Computational Linguistics, Sofia, Bulgaria, 2013, pp. 162–166.

[3] A.E. Balakian, *Dadaism*, in *The Princeton Encyclopedia of Poetry and Poetics*, R. Greene, S. Cushman, C. Cavanagh, J. Ramazani, and P. Rouzer, eds., Princeton University Press, Princeton, 1974, p. 180.

[4] G. Barbieri, F. Pachet, P. Roy, and M.D. Esposti, *Markov constraints for generating lyrics with style*, in *Proceedings of the 20th European Conference on Artificial Intelligence*, L. De Raedt, C. Bessiere, D. Dubois, P. Doherty, and P. Frasconi, eds., IOS Press, Montpelier, France, 2012, pp. 115–120.

[5] M.C. Beardsley, *On the creation of art*, J. Aesthetics Art Criticism 23 (1965), pp. 291–304.

[6] R. Bhatnagar, *Pentametron*, 2012. Available at http://pentametron.com, accessed 9 December 2015.

[7] O. Bown, *Player responses to a live algorithm: Conceptualising computational creativity without recourse to human comparisons?* in *Proceedings of the Sixth International Conference on Computational Creativity*, H. Toivonen, ed., Brigham Young University, Park City, Utah, 2015, pp. 126–133.

[8] J. Carpenter, *Public override void*, 2004. Available at https://slought.org/resources/public_override_void, accessed 9 December 2015.

[9] J. Carpenter, *etc4*, 2007, blog post. Available at http://theprostheticimagination.blogspot.ca/2007/07/etc4.html.

[10] S. Colton, J. Goodwin, and T. Veale, *Full face poetry generation*, in *Proceedings of the Third International Conference on Computational Creativity*, M.L. Maher, ed., computationalcreativity.net, Dublin, Ireland, 2012, pp. 95–102.

[11] P. Dahlstedt, *Between material and ideas: A process-based spatial model of artistic creativity*, in *Computers and Creativity*, J. McCormack and M. d'Inverno, eds., Springer, Berlin, 2012, pp. 205–233.

[12] J. Douglass, *Numeracy and electronic poetry*, J. Math. Arts 8 (2014), pp. 13–23.

[13] A. Eigenfeldt, *Generative music for live musicians: An unnatural selection*, in *Proceedings of the Sixth International Conference on Computational Creativity*, H. Toivonen, ed., Brigham Young University, Park City, Utah, 2015, pp. 142–149.

[14] E. Addad, A.T. Donovan, E. Elshtain, eRoGK7, C. Hardy, illiterate1, JYNX, Nossidge, N.K. Smith, J. Wilberg, *Gnoetry daily*. Available at https://gnoetrydaily.wordpress.com, accessed 9 December 2015.

[15] C.T. Funkhouser, *Prehistoric digital poetry: An archaeology of forms, 1959-1995*, University Alabama Press, Tuscaloosa, Alabama, 2007.

[16] C.T. Funkhouser, *New directions in digital poetry*, A&C Black, London, UK, 2012.

[17] P. Gervás, *An expert system for the composition of formal Spanish poetry*, Knowl.-Based Syst. 14 (2001), pp. 181–188.

[18] P. Gervás, *Exploring quantitative evaluations of the creativity of automatic poets*, in *Workshop on Creative Systems, Approaches to Creativity in Artificial Intelligence and Cognitive Science, 15th European Conference on Artificial Intelligence*, F. van Harmelen, ed., IOS Press, Lyon, France, 2002.

[19] P. Gervás, *Evolutionary elaboration of daily news as a poetic stanza*, in *Proceedings of the IX Congreso Español de Metaheurísticas, Algoritmos Evolutivos y Bioinspirados-MAEB*, F. Herrera and J.A. Gámez, eds., 2013, Salamanca, Spain, pp. 229–238.

[20] P. Gervás, *Constrained creation of poetic forms during theme-driven exploration of a domain defined by an n-gram model*, Connection Sci. 28 (2016), pp. 111–130.

[21] M. Ghazvininejad, X. Shi, Y. Choi, and K. Knight, *Generating topical poetry*, in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, K. Duh and X. Carreras, eds., Association for Computational Linguistics, Austin, Texas, 2016, pp. 1183–1191.

[22] V.P. Glăveanu, *Creativity as a sociocultural act*, J. Creative Behav. 49 (2015), pp. 165–180.

[23] K. Goldsmith, *Flarf is Dionysus. Conceptual Writing is Apollo*, Poetry Foundation, Chicago, 2009.

[24] H. Gonçalo Oliveira, *Tra-la-lyrics 2.0: Automatic generation of song lyrics on a semantic domain*, J. Artif. Gen. Intell. 6 (2015), pp. 87–110.

[25] R. Goodwin, *Adventures in narrated reality*, 2016. Available at https://medium.com/artists-and-machine-intelligence/adventures-in-narrated-reality-6516ff395ba3, retrieved 29 November 2016.

[26] N. Gordon, *Unicorn believers don't declare fatwas*, Poetry Foundation, Chicago, 2009.

[27] J. Harris, *Times haiku: Serendipitous poetry from the New York Times,* 2013-present. Available at http://haiku.nytimes.com/, accessed 9 December 2015.

[28] J. He, M. Zhou, and L. Jiang, *Generating Chinese classical poems with statistical machine translation models*, in *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*, J. Hoffmann and B. Selman, eds., AAAI Publications, 2012.

[29] E. Heep and A. Kapur, *Extracting visual information to generate sonic art installation and performance*, in *Proceedings of the 21st International Symposium on Electronic Art*, P. Pasquier and T. Schiphorst, eds., Vancouver, Canada, 2015.

[30] A. Jordanous, *Four PPPPerspectives on computational creativity*, in *The AISB15's 2nd International Symposium on Computational Creativity (CC2015)*, M.M. al-Rifaie and J. Gow, eds., Society for the Study for Artificial Inteligence, Bath, UK, 2015, p. 16.

[31] A. Kantosalo, J.M. Toivanen, P. Xiao, and H. Toivonen, *From isolation to involvement: Adapting machine creativity software to support human-computer co-creation*, in *Proceedings of the Fifth International Conference on Computational Creativity*, S. Colton, D. Ventura, N. Lavrac, and M. Cook, eds., Association for the Advancement of Artificial Intelligence, Toronto, Canada, computationalcreativity.net, Ljubljana, Slovenia, 2014, pp. 1–7.

[32] A. Karpathy, *The unreasonable effectiveness of recurrent neural networks* (2015), qtd. in (Goodwin, 2016).

[33] J.C. Kaufman and J. Baer, *Beyond new and appropriate: Who decides what is creative?* Creativity Res. J. 24 (2012), pp. 83–91.

[34] A. Knowles and J. Tenney, *A sheet from 'The House', a computer poem* (1968), qtd. in (Funkhouser, 2007).

[35] C.E. Lamb, D.G. Brown, and C.L. Clarke, *Can human assistance improve a computational poet?* in *Proceedings of BRIDGES*, K. Delp, ed., The Bridges Organization, Baltimore, Maryland, 2015, pp. 37–44.

[36] A. Lingentfelter, *Personal communication*, 2017.

[37] T. Lutz, *Stochastiche texte*, Augenblick 4 (1959), pp. 3–9, qtd. in (Roque, 2011); translated to English by Helen MacCormack, 2005.

[38] E. Malmi, P. Takala, H. Toivonen, T. Raiko, and A. Gionis, *DopeLearning: a computational approach to rap lyrics generation*, preprint (2015). Available at arXiv:1505.04771.

[39] H. Manurung, G. Ritchie, and H. Thompson, *Towards a computational model of poetry generation*, Tech. Rep., The University of Edinburgh, Edinburgh, UK, 2000.

[40] H.M. Manurung, *Chart generation of rhythm-patterned text*, in *Proceedings of the First International Workshop on Literature in Cognition and Computers*, 1999, pp. 15–19.

[41] R. Manurung, G. Ritchie, and H. Thompson, *Using genetic algorithms to create meaningful poetic text*, J. Exp. Theor. Artif. Intell. 24 (2012), pp. 43–64.

[42] K. McDowell, *Music, art & machine intelligence 2016 conference proceedings*, 2016. Available at https://medium.com/artists-and-machine-intelligence/music-art-machine-intelligence-2016-conference-proceedings-ea376a4e2576, retrieved 2 December 2016.

[43] N. Montfort and S. Strickland, *Sea and spar between*, Dear Navigator 2, 2010.

[44] A. Mordvintsev, C. Olah, and M. Tyka, *Inceptionism: going deeper into neural networks*, Google Res. Blog 20 (2015).

[45] J. Morris, *Haiku—at random* (1973), qtd. in (Funkhouser, 2007).

[46] Y. Netzer, D. Gabay, Y. Goldberg, and M. Elhadad, *Gaiku: generating haiku with word associations norms*, in *Proceedings of the Workshop on Computational Approaches to Linguistic Creativity*, Association for Computational Linguistics, 2009, pp. 32–39.

[47] S. Opolka, P. Obermeier, and T. Schaub, *Automatic genre-dependent composition using answer set programming*, in *Proceedings of the 21st International Symposium on Electronic Art*, P. Pasquier and T. Schiphorst, eds., Vancouver, Canada, 2015.

[48] F. Pachet and P. Roy, *Markov constraints: Steerable generation of Markov sequences*, Constraints 16 (2011), pp. 148–172.

[49] F. Pachet, P. Roy, J. Moreira, and M. d'Inverno, *Reflexive loopers for solo musical improvisation*, in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, P. Baudisch and M. Beaudouin-Lafon, eds., ACM, Paris, France, 2013, pp. 2205–2208.

[50] A. Papadopoulos, P. Roy, and F. Pachet, *Assisted lead sheet composition using flowcomposer*, in *Proceedings of the International Conference on Principles and Practice of Constraint Programming*, Springer, 2016, pp. 769–785.

[51] G. Percival, S. Fukayama, and M. Goto, *Song2Quartet: A system for generating string quartet cover songs from polyphonic audio of popular music*, in *Proceedings of the 16th ISMIR Conference*, The International Society of Music Information Retrieval, 2015, pp. 114–120.

[52] A. Ramakrishnan and S.L. Devi, *An alternate approach towards meaningful lyric generation in Tamil*, in *Proceedings of the NAACL HLT Second Workshop on Computational Approaches to Linguistic Creativity*, J. Burnstein, M. Harper, and G. Penn, eds., Association for Computational Linguistics, Los Angeles, CA, 2010, pp. 31–39.

[53] M. Rhodes, *An analysis of creativity*, Phi Delta Kappan 42 (1961), pp. 305–310.

[54] A. Roque, *Language technology enables a poetics of interactive generation*, J. Electr. Publ. 14 (2011).

[55] E. Sadler-Smith, *Wallas' four-stage model of the creative process: More than meets the eye?* Creativity Res. J. 27 (2015), pp. 342–352.

[56] M. Scirea, P. Eklund, and J. Togelius, *Toward a context sensitive music generator for affective state expression*, in *Proceedings of the Sixth International Conference on Computational Creativity*, H. Toivonen, ed., late-breaking abstract, Brigham Young University, Association for Computational Creativity, 2015.

[57] V.W. Soo, T.Y. Lai, K.J. Wu, and Y.P. Hsu, *Generate modern style Chinese poems based on common sense and evolutionary computation*, in *Proceedings of the Conference on Technologies and Applications of Artificial Intelligence (TAAI)*, C.K. Ting, S.J. Yen, Y.G. Wu, M.F. Horng, and C.L. Chen, eds., IEEE, Taiwan, ROC, 2015, pp. 315–322.

[58] C. Stross, *Lovecraft.pl* (2013), blog post. Available at http://www.antipope.org/charlie/blog-static/2013/12/lovebiblepl.html, accessed 21 December 2016.

[59] J. Toivanen, H. Toivonen, A. Valitutti, and O. Gross, *Corpus-based generation of content and form in poetry*, in *Proceedings of the Third International Conference on Computational Creativity*, M.L. Maher, ed., computationalcreativity.net, Dublin, Ireland, 2012, pp. 175–179.

[60] J.M. Toivanen, O. Gross, and H. Toivonen, *The officer is taller than you, who race yourself! Using document specific word associations in poetry generation*, in *Proceedings of the Fifth International Conference on Computational Creativity*, computationalcreativity.net, 2014, pp. 355–359.

[61] J.M. Toivanen, M. Järvisalo, and H. Toivonen, *Harnessing constraint programming for poetry composition*, in *Proceedings of the Fourth International Conference on Computational Creativity*, M.L. Maher, ed., computationalcreativity.net, Sydney, Australia, 2013, pp. 160–167.

[62] University of Helsinki Computer Science Department, *Works of artistic nature*. Available at https://www.cs.helsinki.fi/discovery/works-artistic-nature, accessed 6 December 2016.

[63] T. Veale and Y. Hao, *Exploiting readymades in linguistic creativity: A system demonstration of the jigsaw bard*, in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Systems Demonstrations*, D. Lin, ed., Association for Computational Linguistics, Portland, Oregon, 2011, pp. 14–19.

[64] D. Ventura, *Mere generation: Essential barometer or dated concept?* in *Proceedings of the Seventh International Conference on Computational Creativity*, F. Pachet, A. Cardoso, V. Corruble, and F. Ghedini, eds., Sony CSL, Paris, 2016, pp. 17–24.

[65] Q. Wang, T. Luo, D. Wang, and C. Xing, *Chinese song iambics generation with neural attention-based model*, preprint (2016). Available at arXiv:1604.06274.

[66] Z. Wang, W. He, H. Wu, H. Wu, W. Li, H. Wang, and E. Chen, *Chinese poetry generation with planning based neural network*, preprint (2016). Available at arXiv:1610.09889.

[67] T.B. Ward, S.M. Smith, and R.A. Finke, *Creative cognition*, in *Handbook of Creativity*, R.J. Sternberg, ed., Cambridge University Press, Cambridge, 1999, pp. 189–212.

[68] R. Yan, *i, Poet: Automatic poetry composition through recurrent neural networks with iterative polishing schema*, in *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, S. Kambhampati, ed., AAAI Press, New York, 2016.

[69] R. Yan, H. Jiang, M. Lapata, S.D. Lin, X. Lv, and X. Li, *i, Poet: Aautomatic Chinese poetry composition through a generative summarization framework under constrained optimization*, in *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*, F. Rossi, ed., AAAI Press, Beijing, China, 2013, pp. 2197–2203.

[70] V. Zappi and A. McPherson, *The D-Box: How to rethink a digital musical instrument*, in *Proceedings of the 21st International Symposium on Electronic Art*, P. Pasquier and T. Schiphorst, eds., Vancouver, Canada, 2015.

[71] M.H. Zareei, D.A. Carnegie, and A. Kapur, *Noise square: Physical sonification of cellular automata through mechatronic sound-sculpture*, in *Proceedings of the 21st International Symposium on Electronic Art*, P. Pasquier and T. Schiphorst, eds., Vancouver, Canada, 2015.

[72] X. Zhang and M. Lapata, *Chinese poetry generation with recurrent neural networks*, in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, A. Moschitti, B. Pang, and W. Daelemans, eds., Association for Computational Linguistics, Doha, Qatar, 2014, pp. 670–680.