TwitSong 3.0: Towards Semantic Revisions in Computational Poetry

Carolyn Lamb and Daniel G. Brown
David R. Cheriton School of Computer Science
University of Waterloo
Waterloo, Ontario, Canada
c2lamb@uwaterloo.ca; dan.brown@uwaterloo.ca

Abstract

We present TwitSong 3.0, a found poetry system which locates candidate lines in a source text or generates them, then edits the lines repeatedly to increase their score on measures of meter, topicality, imagery, and emotion. We evaluate TwitSong 3.0 using a survey of domain experts. The system's editing process does significantly improve its lines, although the resulting poems are not always coherent, and the experimental evidence suggests that the improvement may not occur through the precise mechanisms we intended.

Introduction

This paper continues the work of poetry generation begun in our previous two papers (Lamb, Brown, and Clarke 2015; Lamb, Brown, and Clarke 2017), in which we use our TwitSong system to generate found poetry based on the news. TwitSong works by gathering candidate lines from a large topical corpus, rating them on various scales for their poetic suitability, and assembling them into a formal, rhymed poem. Prior versions of TwitSong produced mixed results; our motivation in the current study was to improve the system's foundations by making its underlying process more sophisticated.

The current generation of TwitSong introduces a new mechanism. TwitSong 3.0 is able to make targeted, goal-directed edits to its own work using our Editorial Algorithm, a form of genetic programming inspired by the human creative process. This builds on the work of Gervás (2016), which also edits candidate lines for traits such as rhyme and number of syllables. TwitSong 3.0 goes further by editing for semantic traits such as topicality and emotion.

We evaluate TwitSong 3.0 and find that using the Editorial Algorithm significantly improves the resulting poems in terms of expert judges' pairwise preferences. However, this improvement is small and much of it is due to the Editorial Algorithm's effect on meter. Alternatively, the improvement can be interpreted as a result of contradictory effects in our line selection criteria. While the overall quality of the poems is not what we hoped, TwitSong 3.0's evaluation also serves as an example of good practice for computational poetry evaluation, and as an application of the evaluation principles we have developed in our prior work (Lamb, Brown, and Clarke 2016; Lamb, Brown, and Clarke 2018).

Related Work

TwitSong (Lamb, Brown, and Clarke 2015; Lamb, Brown, and Clarke 2017) is a found poetry system in which human-written candidate lines are modified and recombined. Similar found poetry systems in the computational creativity literature include The Poet's Little Helper (Astigarraga et al. 2017) and DopeLearning (Malmi et al. 2015). Generating poetry based on the news is a poetic goal previously worked for by systems including P.O.Eticus (Toivanen, Gross, and Toivonen 2014) and Pemuisi (Rashel and Manurung 2014). Evaluation of earlier versions of TwitSong showed that rating lines on criteria such as topicality and emotion could produce overall better poems than a control (Lamb, Brown, and Clarke 2015), but that automating these ratings did not always produce the desired effect (Lamb, Brown, and Clarke 2017).

The idea of creativity as a generation-evaluation loop, with a creator able to repeatedly evaluate its unfinished work and make targeted improvements, is important in many theories of computational creativity and creativity psychology (Ward, Smith, and Finke 1999; García et al. 2006; Simonton 2011; Dahlstedt 2012). The use of looping, targeted edits for poetry specifically was previously done by Diaz-Agudo *et al.* (Díaz-Agudo, Gervás, and González-Calero 2002) with the COLIBRI system, and continued by Gervás with WASP (Gervás 2013a; Gervás 2013b; Gervás 2016). Various versions of these systems edit candidate lines for rhyme, meter, stress pattern, excessive similarity to the source text, excessive repetition, sentence length, verse length, and grammatical plausibility of the final word in the sentence. Gervás expresses a desire to use similar techniques to optimize for semantic traits, such as topicality (Gervás 2016), but no such method has yet been found. Apart from being included for publication in a book about computational poetry (Gervás 2013a), neither COLIBRI nor WASP have been formally evaluated.

Formal evaluation for computational creativity systems is a topic worthy of books in itself; our previous survey (Lamb, Brown, and Clarke 2018) gives a detailed interdisciplinary overview. Many computational poetry systems are not formally evaluated, or are evaluated without adhering to what we argue are best practices for evaluation. In this paper we focus on a few such best practices: the testing of falsifiable hypotheses about a system's creative output; the use

of domain expert judges; and the use of domain-specific, evidence-based testing criteria.

Our own prior research (Lamb, Brown, and Clarke 2016), analyzing the written responses of poetry quasi-experts to examples of computational poems, identified four major criteria that such experts look for when expressing their opinions about poetry. These criteria are Reaction (the expert's personal, emotional response to the poem), Meaning (the sense that the poem conveys a message), Novelty (the sense that the poem says something different from what has been said before), and Craft (the skill and technique with which the poem is constructed, including specific poetic devices). Each of these criteria is also divided into subcriteria. To our knowledge these are the first poetry evaluation criteria that have been developed directly from the study of poetry experts' responses, rather than stated ad hoc by the researcher or lifted from another creative domain. More work remains to be done on the four poetry criteria before they constitute a reliable and valid set of constructs for testing, but the same can be said of any other existing group of criteria, so for now we refer to the four criteria throughout our own research.

How TwitSong 3.0 works

For line representation, RhymeSet construction, line judging, and poem construction, TwitSong 3.0 is built on similar code to its previous generations (Lamb, Brown, and Clarke 2015; Lamb, Brown, and Clarke 2017). In brief, a source text is mined for potentially rhyming phrases of the appropriate length, and these phrases are grouped based on end rhyme. Syllabification and rhyme detection is performed using the CMU Pronouncing Dictionary and with Hirjee and Brown's Rhyme Analyzer code (Hirjee and Brown 2010). These representation, judgment, and construction mechanisms are not new; what is new in this generation is the Editorial Algorithm and its Markov chain-based targeted reconstruction of lines.

Each candidate line in a RhymeSet is given an automated score based on our line judgment criteria. For TwitSong 3.0, these are:

- Emotion. A target emotion is chosen for the poem based on prevalence in the source text and appropriateness for the topic. The line is then scored by adding together the scores of each of its individual words for this emotion in the NRC Hashtag Emotion Lexicon (Mohammad and Kiritchenko 2015), which was specifically developed for use with short texts like tweets or the lines of a poem. The goal with measuring Emotion is to produce an appropriate emotional reaction in the reader, for the Reaction criterion.
- Imagery. Each line is given a score for the concreteness of its imagery by adding together the scores of its individual words in the Regressive Imagery Dictionary (Provalis 1990), a dictionary used by Simonton (Simonton 1990) when statistically comparing more and less successful poems. "Primary process" words in the dictionary are associated with more concrete imagery and more successful poetry overall. Kao and Jurafsky (Kao and Jurafsky 2012), analyzing professional and amateur contemporary poetry with a similar tool, also found that concrete imagery was one of the strongest predictors of more successful professional poetry. Imagery is itself a subcriterion of the Craft criterion in our research, and given its importance in other poetry research, we posit that it is one of the most important such factors.
- Meter. Each line is given a score between 0 and 1 based on its adherence to an iambic metrical scheme: for a score of 1, all even numbered syllables should be stressed and all odd numbered syllables should be unstressed. Because the stresses of single-syllable words can be difficult to discern, and because the CMU pronouncing dictionary also includes secondary stresses, our Meter scores are not exact. For metrical poetry, this is another obvious subset of Craft, and is necessary in order to produce poems of the desired form. The pre-selection of lines of the appropriate length and with appropriate rhymed endings also constitutes Craft.
- Topicality. Source texts are chosen for their general relevance to a specified topic. Each line is divided into trigrams based on a sliding window, and lines are given a higher score if they contain trigrams which occur more frequently in the data set. Early experiments with this version of TwitSong showed that the trigram frequency measure selected for common and intelligible turns of phrase and weeded out nonsensical combinations, but it often also resulted in the selection of bland lines which did not make it clear what the poem was about. So a specificity measure was added: the 30 most topical words in a given data file are selected by dividing the frequency of each word in the source text by its frequency in a non-topical comparison text (in this case, the comparison text is a compilation of poems from Poetry Magazine; a factor is added to the frequency to prevent division by zero). A trigram containing one or more of these most topical words receives a bonus to its topicality score. Topicality is necessary for the criterion of Meaning.

. The four automated scores are then normalized and summed to give a line's total score. A RhymeSet is given its own score based on the average of the two top scoring lines in the set, and the top two lines of the highest scoring RhymeSets are arranged into a metrical rhyming poem by the poem construction mechanism.

TwitSong 3.0 uses sets of news articles as its source texts and assembles its lines into quatrains in Common Meter—an ABAB rhyme scheme with four iambs (eight syllables) in the A lines and three iambs (six syllables) in the B lines. This is the form of many hymns, including "Amazing Grace," as well as other popular poems and songs. This is different from how poems were constructed in previous versions of TwitSong. However, the major change in TwitSong 3.0 is the introduction of the Editorial Algorithm, by which the most promising lines can be refined by TwitSong after they are selected.

The Editorial Algorithm

The Editorial Algorithm is a form of genetic algorithm. However, instead of randomly recombining the most suc-

cessful candidates in each generation—a technique which bears little resemblance to how human poets revise their work—we use a targeted edit at each step, replacing the words in each line that contribute most to the line's worst-performing metric, out of the four metrics of Topicality, Emotion, Imagery, and Meter. We detail this algorithm below.

1. Initialization. The source text is read and the dictionaries used for each criterion are initialized, including the trigram frequency dictionary and identification of most topical words. We also initialize an interpolated Markov model (Salzberg et al. 1999) which can be used to generate additional text in the style of the source text. The Markov model can be up to order 3, but can flexibly reduce its order. If the model generates no results, or only one result, for a 3-gram, then it reduces the 3-gram to a 2-gram or 1-gram. The Markov model is trained to recognize punctuation that could indicate the end of a sentence or line, and runs until it generates an "end of line" marker; in an earlier version that did not use these markers, it was too common for a line to end on a preposition or other unsuitable word. Because the "end of line" marker is not guaranteed to occur after exactly 6 or 8 syllables, the Markov model in practice is run repeatedly until it generates a line that happens to be of the right length.
2. Line initialization. The source text is divided into lines of 6 or 8 syllables, separated by punctuation such as periods, question marks, colons, and commas. For each of the 30 most topical words, the Markov model generates a special line by starting with the listed word and iterating until it has a line with the appropriate number of syllables. Both the source lines and the Markov lines are then sorted into RhymeSets based on their end rhymes.
3. Scoring. The lines in each RhymeSet are scored based on the four metrics, and each RhymeSet is scored based on its best two lines. RhymeSets with a single line are scored, but penalized.
4. Trimming. For each RhymeSet, any line that is identical to the RhymeSet's top scoring line, or that begins or ends with an identical word, is removed. Optionally, the programmer can also specify removal words that can only appear once in each RhymeSet; if the top scoring line contains one of these words, then any other line containing that word is removed. This is useful for preventing repetition. If more than 15 lines remain in the RhymeSet, it is then trimmed down to only its 15 highest scoring lines. The RhymeSets are re-scored and the 50 highest scoring RhymeSets are kept for the next generation, with RhymeSets of only a single line being removed first.
5. Edit planning. This is where the Editorial Algorithm identifies which words most need to be replaced. Each line in each RhymeSet is analyzed based on the four criteria. The criterion with the lowest normalized score, as well as any other criterion which is under a certain threshold, is selected for analysis. Each word in the line is then inspected for its contribution to this criterion, and the lowest performing word is selected for replacement. ("Stop words," such as "the" and "of," are not excluded from this process; the thinking is that, if a stop word is present, there is no *a priori* reason why an alternate version of the line might not use a different sentence structure and have a higher-scoring word there instead.) For example, if Imagery is selected, then words that are very abstract are selected. We detail this process further below
6. Word replacement. The selected lines are sent to the Markov model which generates candidate replacement lines, starting with the selected underperforming word and replacing it and all subsequent words. (An earlier prototype of TwitSong replaced only the underperforming word, but this led to choppy and repetitive lines; an example is given in Table 1.) Because there is no guarantee that the replacement words will actually be better, the Markov chain generates many candidate replacement lines—20 for each selected starting word. These are then assigned to appropriate RhymeSets.
7. Successive generations. TwitSong repeats steps 3 through 6 to a maximum of 100 generations, or until the average score of the best ten lines stops increasing. In practice, the program very rarely runs for more than 15-20 generations, and sometimes as few as 3.
8. Poem construction. The top two lines each from the two highest scoring RhymeSets are selected. These are arranged into a quatrain in Common Meter.
9. Title generation. TwitSong generates a title for each of its poems, but the title generation mechanism is separate from the rest of the Editorial Algorithm. During the Line Initialization step, in addition to creating the initial RhymeSets, TwitSong also gathers a set of lines from the source text of 3 to 5 syllables without grouping them into RhymeSets. These potential title lines are then scored based on the four combined metrics and checked against the list of most topical words. Ideally, lines containing the first most topical word are selected and the highest scoring such line becomes the title. If there are no such lines, TwitSong will iterate down the list of most topical words. If no potential title line contains any of the 30 most topical words, TwitSong will choose the overall highest scoring potential title.

let wall street start off wall detroit's
and wall street start wall voiced
let wall street start wall street wall point
wall street out loud wall point

Figure 1: An early example of a poem from a prototype Editorial Algorithm, using Bernie Sanders' lines from presidential debate transcripts as a source text. In this prototype, pairs of words were replaced during each edit. (An even earlier version, replacing single words, resulted in lines like "let wall wall wall wall wall wall street".) This problem was avoided by a later protocol in which the target word and everything after it in the line is re-generated at once.

Source Texts

TwitSong 3.0's architecture allows it to generate poems quickly. In particular, the use of a Markov chain means that a relatively small source text can be used to generate poems. TwitSong 3.0's lower limit, before it stops being able to come up with sufficient numbers of rhymes for a quatrain, seems to be around 20 kilobytes of text. Therefore, it can be initialized with only a handful of articles on a breaking news topic.

We generated a great number of poems using TwitSong 3.0, mostly based on news articles from the BBC [1], CBC [2], Maclean's [3], and The Guardian [4]. We chose these sources because they are mainstream, professional English language news sources which operate without a paywall. Occasionally we veered into other sources. For instance, when blockbuster movies were released, we collected fan responses to the movies from Tor.com [5] and The Mary Sue [6]. We also tried alternative, non-news sources for some poems, such as classic novels available on Project Gutenberg[7].

The Evolutionary Algorithm in action

As an illustration, we show how the Evolutionary Algorithm uses its word replacement techniques on lines for a poem about the film *Avengers: Infinity War*.

One of the starting lines for this poem is:

thanos to grow the universe

This line receives high scores for meter and imagery, but a low score for topicality and a moderately low score for the chosen emotion, surprise. As both topicality and emotion are below their minimum thresholds, the Editorial Algorithm focuses on both of these.

Since topicality is calculated based on trigrams, TwitSong splits this line into its component trigrams:

thanos to grow / to grow the / grow the universe

The first and last trigrams are selected because they are not found in the trigram dictionary. Thus, TwitSong generates a set of candidate replacement lines starting at the beginning of the line, and a set of candidate replacement lines modifying only the last three words.

For emotion, TwitSong splits the line into its component words:

thanos / to / grow / the / universe

None of these individual words are very associated with the emotion of surprise, and some do not appear in the lexicon. Therefore, TwitSong flags all of them, and generates a maximal set of candidate replacement lines (a different set beginning the word replacement at each word).

The completed poem from this run of TwitSong reads:

marvel had the fall of your mouth
luke of this journey through
infinity stone to point out
gags to where thor is too

[1] http://bbc.com/news
[2] http://www.cbc.ca/news
[3] http://www.macleans.ca
[4] https://www.theguardian.com/international
[5] https://www.tor.com/
[6] https://www.themarysue.com/
[7] http://www.gutenberg.org/

Group A
FOR CANADA *(Olympics, joy)* hamelin pointing at the world team made it would be fair swiss stones for pavel is absurd swiss stones for him and there
Group B
WHY IS TRUMP SILENT *(Mueller investigation, disgust)* republican claims he will do flynn pleaded not care less committee has to look into pleaded not to the press
Group C
WAKANDA *(Black Panther, trust)* blackness as we love to her aid killmonger's plan to come conflict the atlantic slave trade sword and it was awesome

Figure 2: Example poems from the three experimental groups.

Evaluation

Our goal in evaluating TwitSong was to falsifiably test whether or not the Editorial Algorithm and its associated line rating techniques improved TwitSong's poetry.

Method

We assembled three experimental groups of poems: Group A, Group B, and Group C.

Poems from Group A were generated according to the Editorial Algorithm described above. The best lines of each generation were edited with the goal of increasing their summed score on our criteria of Topicality, Emotion, Imagery, and Meter.

Poems from Group B were generated with a minimal version of the Editorial Algorithm. Lines were taken from a source text and generated based on a Markov chain trained on the source text. If this resulted in enough RhymeSets to produce a quatrain in Common Meter, the program was stopped there. Otherwise, it was allowed to iterate and perform the Editorial Algorithm for *only* enough generations to produce a valid quatrain. Every line was then assigned a score of zero, and the lines for the quatrain were chosen arbitrarily. Group B was meant as a control group in which the Editorial Algorithm did as little to improve the poems as possible, yet the poems were similar to the poems of Group A in every other respect.

We chose this method for our control group rather than using output from previous versions of TwitSong because

Emotion	Frequency	Topics
Disgust	7	Mueller investigation; the Parkland school shooting; Rex Tillerson; Doug Ford's election campaign in Ontario; March For Our Lives; the Stormy Daniels scandal; Viktor Orban's election in Hungary
Fear	6	Winter Olympics (2); Uber self-driving car crash; the Russian election; Austin bombing; NAFTA negotiations; Syrian chemical attack
Anticipation	5	Kim Jong Un's visit to China; Russian spy poisoning; US trade war; North Korea; Michael Cohen warrant
Anger	4	The Cambridge Analytica scandal; Facebook; Tim Hortons; Mark Zuckerberg
Joy	3	Winter Olympics (1); A Wrinkle in Time; Easter on April 1
Sadness	3	Stephen Hawking's death; Good Friday; Humboldt Broncos bus crash
Surprise	1	The Oscars
Trust	1	Black Panther

Table 1: Frequency of emotions from the NRC Hashtag Emotion Lexicon assigned to poems on different topics, from the group of 30 topics that were selected for the study. The topics in this table are sorted by associated emotion for ease of reading, and their order does not correspond to the ordering of topics in the study.

previous versions used different source text (Twitter) and a different metrical form; this is the first time that the TwitSong system has been tested on news. Using a different baseline control group, such as random or human-generated text, would have given insight into where the poems stand in terms of overall quality, but would not have answered our specific experimental question about whether the Editorial Algorithm was improving the poems.

Poems from Group C were generated with a *reversed* Editorial Algorithm. That is to say, the line rating and edit planning steps were programmed to minimize instead of maximizing the poem's scores. So these poems were the Editorial Algorithm's attempt to make poems that were off-topic, unrelated to the selected emotion, abstract / devoid of imagery, and that failed to conform to an iambic stress pattern.

We chose a set of 30 news topics that were current at the time of the study and generated a Group A, Group B, and Group C poem for each. We then constructed a test set for our study in which, for each of the 30 topics, two of the groups were selected. The order of the news topics was not randomized, but the order of pairings (A vs B, A vs C, B vs A, B vs C, C vs A, or C vs B) was randomized across the set of news topics. All eight of the emotions from the NRC Hashtag Emotion Lexicon were present in our set of poems, but we made no attempt to balance or equalize the appearance of different emotions, instead picking the emotion that was most prevalent in articles describing each topic according to the NRC Hashtag Emotion Lexicon, with some normalization and some exceptions (see Table 1 for a full list).

We recruited experimental subjects by snowball sampling in order to include a reasonable number of poetry experts in our analysis; one of our authors is a published poet and recruited their own poetry contacts for the study. Each subject was directed to an online survey in which they were presented with each of the 30 pairs of poems and asked their opinions. Participants were also asked a few demographic questions and given a freeform text box at the end for other comments about the study. The full study took about 40 minutes and participants were given 10 Canadian dollars as remuneration.

The bulk of the survey used a pairwise forced choice paradigm. For each pair of poems, participants were asked the following questions:

- Which poem do you prefer? *(General/Reaction)*
- Which poem is more creative? *(General)*
- Which poem does a better job expressing the emotion of [emotion]? *(Reaction)*
- Which poem does a better job describing the topic of [topic]? *(Meaning)*
- Which poem is more new and different? *(Novelty)*
- Which poem has better imagery? *(Craft)*

In a previous study (Lamb, Brown, and Clarke 2017) we included a question about cohesiveness. As TwitSong 3.0 does not contain mechanisms specifically designed to increase cohesiveness, we omitted this question from our study. Given the way many participants ended up focusing on the poems' lack of cohesion, this may have been a mistake.

Results

Demographics We divided our survey participants into experts and non-experts based on their self-reported experience with poetry. Experts were defined as participants whose poetry had been published in a magazine, anthology, collection, etc.

32 poetry experts participated in our study. This included 11 men, 10 women, 9 non-binary poets, and two experts who did not disclose their gender. (This is probably a serious overrepresentation of non-binary poets, but we do not expect it to affect our study results as none of the poems in the sample discuss queer/trans* issues.) Their ages ranged from 22 to 57, averaging 38. 28 of the 32 experts were native English speakers.

49 non-experts participated in our study, including 17 men, 27 women, and 5 non-binary participants. Their ages ranged from 17 to 64, averaging 32. 37 of the 49 non-experts were native English speakers.

We observed high attrition as the survey was rather long. Only 18 experts and 28 non-experts managed to complete every question. However, since the order of appearance of poems from different groups was randomized, this still left us with a good number of pairwise comparisons and did not present a major statistical problem.

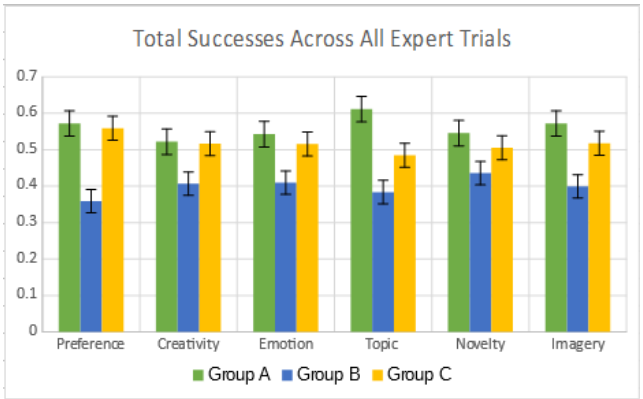

Figure 3: Success rates for types of computationally generated poems in pairwise comparisons with other poems, judged by experts. The height of a given bar represents the number of times a poem from that category was selected in preference to any other poem. The groups of bars add up to 150% because, for each category, the $\frac{1}{3}$ of trials in which a poem from that category does not appear are not considered. Error bars represent 95% confidence intervals, prior to Bonferroni correction.

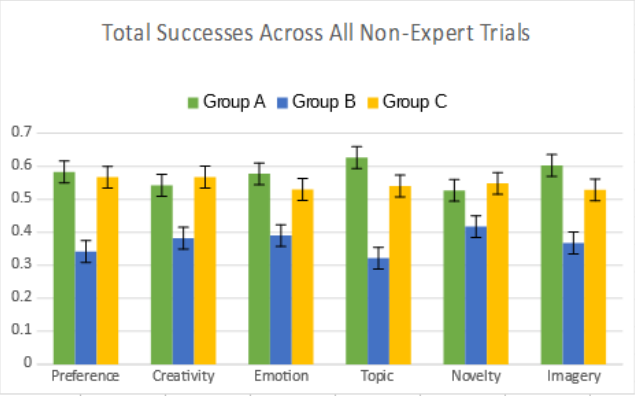

Figure 4: Success rates for types of computationally generated poems in pairwise comparisons with other poems, judged by non-experts. The height of a given bar represents the number of times a poem from that category was selected in preference to any other poem. Error bars represent 95% confidence intervals, prior to Bonferroni correction.

Group comparison We evaluated pairwise preferences between poems by treating them as a binomial distribution; statistical significance is calculated using the binomial theorem for cumulative probability. The null hypothesis is that the probability of choosing a poem from one group over a poem from another, on any question, is 50%. As there are six questions, we applied a Bonferroni correction for multiple hypotheses, resulting in an alpha level of .0083 per test.

Our results are shown in Figures 3 and 4. As we had hoped, experts significantly preferred poems from Group A to poems from Group B on all six questions,$p < 0.0083$ for all. The differences between Groups B and C were not significant; surprisingly, neither were any differences between groups A and C.

Non-experts, like experts, significantly preferred poems from Group A to poems from Group B, $p < 0.0083$ for all questions. They also significantly preferred Group C to Group B on all questions, $p < 0.0083$. The differences between Groups A and C were not significant for non-experts.

Rather than the expected $A > B > C$ hierarchy, there is little difference between A and C. For experts there is some evidence of a possible $A > C > B$ ordering, but with the differences other than $A > B$ too slight to be significant. For non-experts, A and C seem to be genuinely statistically the same. This is illustrated in Figures 3 and 4.

Correlation between questions We looked at the correlations between the answers to our different questions, to see if our questions were truly capturing different dimensions underlying Product creativity. The results, for experts, are in Table 2. All the correlations between questions are above 0, which is not worrisome, since it is expected that a preference for a poem in some questions would have a priming effect on the other questions. However, some correlations are weak to moderate, while others are strong. There is a notable gap between the strongest moderate correlation (preference and emotion, Pearson's r=0.56) and the weakest strong correlation (creativity and imagery, r=0.84).

It appears that the measures of preference, creativity, novelty, and imagery are all strongly intercorrelated, while emotion and topic are more independent. This implies that experts evaluate the poems on three basic dimensions. One is how well the poem represents the target topic; another is how well the poem expresses the target emotion; a third is a more nebulous measure of how "good" the poem is, including novelty, imagery, and overall preference. Non-experts exhibited the same pattern as experts, with three underlying dimensions.

Freeform comments We counted and categorized the freeform comments made by experts and non-experts. Experts commented more often than non-experts, but there was more unity in the types of comments made by non-experts.

Several experts and non-experts stated that the poems didn't represent the intended emotions very well. Both experts and non-experts wished that there was a neutral/none/both option for times when neither poem met its targets well.

Experts were concerned about the poems' coherence. Several stated that the poems were incoherent, or that they cared more about coherence than the items the survey asked for. Two experts added that some lines were great, but that they were spoiled by proximity to incongruous or "word salad" lines.

	Preference	Creativity	Emotion	Topic	Novelty	Imagery
Preference	1					
Creativity	0.855	1				
Emotion	0.563	0.360	1			
Topic	0.525	0.320	0.344	1		
Novelty	0.839	0.861	0.351	0.227	1	
Imagery	0.904	0.837	0.421	0.423	0.892	1

Table 2: Correlations (Pearson's R) between answers to each of the six questions, as judged by experts.

Non-experts made more comments about the overall quality of the poems, although they were divided in their responses. Several said that the overall set of poems, or the idea for the study, was interesting or cool. Some said that the poems overall are not very good, while others said that some individual poems were quite good. Several non-experts indicated that the poems were hard to understand or didn't make sense, which may be the non-expert version of complaints about coherence.

One expert commented, "My god, that was awful. The poems were some of the worst computer-generated texts I've ever seen." In contrast, a non-expert said, "This is a really interesting study–I was trying to guess which poems were computer-generated as I did the survey, and I couldn't tell most of the time!" This comment is notable since we had intended it to be clear that *all* poems in the study were computer-generated.

Discussion

We were surprised by our results. It seems that the Evolutionary Algorithm improves poems even when told to make the poems worse.

We can think of a few ways to interpret this result. One is that our line rating metrics are useless and something else about the Evolutionary Algorithm improves the poems. However, we are not sure what this would be. Although participants claimed not to see much difference between the groups, they detected a statistically significant difference. This difference must be due to the line rating metrics and their use, as there were no other consistent differences between the poems from the three groups.

It is possible that, while the line rating metrics are useful, their reverse versions are also useful. This is most easily explained with Meter. A line with a score of 1.0 for meter is a perfect iambic line. However, the opposite of an iambic line is not an unmetrical line. Instead, the opposite of an iambic line is a trochaic line. It is very likely that, while lines from Group B had random stress patterns and lines from Group A were mostly iambic, lines from Group C were mostly trochaic. Looking at the poems from group C, many do contain trochaic or close to trochaic meter, with lines like *game that finish gave a doping* or *shooting following his thursday*.

This explanation is speculative due to a flaw in our experiment: we did not include a question like "Which poem has better meter and rhythm?" even though rhythm and meter are valid subcategories of Craft.

If Groups A and C have good meter and Group B does not, then there are two possible explanations for the other results. One is that the answers to the other questions are illusions—survey participants prefer the poems with better meter, and this increases scores in other areas solely due to priming. Another possible explanation is that other line rating metrics also exhibit this reverse effect. A poem with low Topicality might contain more unusual trigrams and, thus, more Novelty. A poem with a low rating for one emotion might end up exhibiting another, equally interesting emotion. Lines with lower Imagery might use more straightforward language and therefore be more coherent. This explanation does not completely explain the data; for instance, it does not explain why Groups A and C are both more topical and more novel than Group B.

We suspect a combination of both explanations. Both Group A and C improve on the Group B poems, especially for meter, but Group A is slightly more on target with regards to its other goals. Experts are more sensitive to this, resulting in a ranking where Group A (slightly, non-significantly, but consistently) outperforms Group C, while non-experts are more fully swayed by meter and less able to perceive other improvements. Although the difference between Groups A and C when judged by experts is not significant, there is only a 1/64 chance that Group A would outperform Group C on all six questions if the data was random.

If this combined explanation is true then we would expect several consequences in further experiments. First, we would expect that, if we did include a question about Meter, Group A and C would prove to have better meter than Group B, and other questions would be highly correlated with Meter, especially for non-experts. Second, if we had a better implementation of our line ratings, then the difference between Group A and Group C, at least for experts, would increase.

Conclusion

TwitSong 3.0 was meant to build on the accomplishments of WASP (Gervás 2013a; Gervás 2013b; Gervás 2016). By making goal-directed edits to candidate lines as WASP does, TwitSong 3.0 measurably improves these lines. However, our goal was to expand these techniques to semantic goals such as topicality and emotion, and our success at this was limited: the improvements that our system made to its lines were mostly in the area of meter. Editing lines to be more topical remains an open problem. Additionally, TwitSong 3.0's poems were generally not very coherent or well liked

by expert judges.

While TwitSong 3.0 has mixed success as a poetry system, it also exemplifies the importance of well-constructed evaluation with expert judges and falsifiable hypotheses. Without such testing, we might have sensed that the generated poetry wasn't as good as we wanted, but we would not have had the detailed statistical insight that helped us figure out the reason for this and to discover one part of the system (meter) that was working well. There is room to improve our evaluation techniques further, for example, by testing and standardizing a more robust set of questions.

References

[Astigarraga et al. 2017] Astigarraga, A.; Martínez-Otzeta, J. M.; Rodriguez, I. R.; Sierra, B.; and Lazkano, E. 2017. Poet's Little Helper: a methodology for computer-based poetry generation. A case study for the Basque language. In *Proceedings of the Workshop on Computational Creativity in Natural Language Generation (CC-NLG 2017)*, 2–10.

[Dahlstedt 2012] Dahlstedt, P. 2012. Between material and ideas: a process-based spatial model of artistic creativity. In *Computers and Creativity*. Berlin: Springer. 205–233.

[Díaz-Agudo, Gervás, and González-Calero 2002] Díaz-Agudo, B.; Gervás, P.; and González-Calero, P. A. 2002. Poetry generation in COLIBRI. In *Advances in Case-Based Reasoning*. Springer. 73–87.

[García et al. 2006] García, R.; Gervás, P.; Hervás, R.; Pérez, R.; and ArÃmbula, F. 2006. A framework for the ER computational creativity model. In *MICAI 2006: Advances in Artificial Intelligence*, 70–80. Apizaco, Mexico: Springer.

[Gervás 2013a] Gervás, P. 2013a. Computational modelling of poetry generation. In *Artificial Intelligence and Poetry Symposium, AISB Convention*. Exeter University, UK: Society for the Study of Artificial Intelligence and the Simulation of Behaviour.

[Gervás 2013b] Gervás, P. 2013b. Evolutionary elaboration of daily news as a poetic stanza. In *Proceedings of the IX Congreso Español de Metaheurísticas, Algoritmos Evolutivos y Bioinspirados-MAEB*, 229–238.

[Gervás 2016] Gervás, P. 2016. Constrained creation of poetic forms during theme-driven exploration of a domain defined by an n-gram model. *Connection Science* 28(2):111–130.

[Hirjee and Brown 2010] Hirjee, H., and Brown, D. G. 2010. Using automated rhyme detection to characterize rhyming style in rap music. *Empirical Musicology Review*.

[Kao and Jurafsky 2012] Kao, J., and Jurafsky, D. 2012. A computational analysis of style, affect, and imagery in contemporary poetry. In *NAACL Workshop on Computational Linguistics for Literature*, 8–17.

[Lamb, Brown, and Clarke 2015] Lamb, C. E.; Brown, D. G.; and Clarke, C. L. 2015. Can human assistance improve a computational poet? *Proceedings of Bridges 2015: Mathematics, Music, Art, Architecture, Culture* 37–44.

[Lamb, Brown, and Clarke 2016] Lamb, C.; Brown, D. G.; and Clarke, C. L. 2016. Evaluating digital poetry: Insights from the CAT. In *Proceedings of the Seventh International Conference on Computational Creativity*. Association for Computational Creativity.

[Lamb, Brown, and Clarke 2017] Lamb, C.; Brown, D.; and Clarke, C. 2017. Incorporating novelty, meaning, reaction and craft into computational poetry: a negative experimental result. In *Proceedings of the Eighth International Conference on Computational Creativity*. Association for Computational Creativity.

[Lamb, Brown, and Clarke 2018] Lamb, C.; Brown, D. G.; and Clarke, C. L. 2018. Evaluating computational creativity: An interdisciplinary tutorial. *ACM Computing Surveys (CSUR)* 51(2):28.

[Malmi et al. 2015] Malmi, E.; Takala, P.; Toivonen, H.; Raiko, T.; and Gionis, A. 2015. DopeLearning: a computational approach to rap lyrics generation. *arXiv preprint arXiv:1505.04771*.

[Mohammad and Kiritchenko 2015] Mohammad, S. M., and Kiritchenko, S. 2015. Using hashtags to capture fine emotion categories from tweets. *Computational Intelligence* 31(2):301–326. Full lexicon available at `http://saifmohammad.com/WebPages/lexicons.html`.

[Provalis 1990] Provalis. 1990. Regressive imagery dictionary. `https://provalisresearch.com/products/content-analysis-software/wordstat-dictionary/regressive-imagery-dictionary/`.

[Rashel and Manurung 2014] Rashel, F., and Manurung, R. 2014. Pemuisi: a constraint satisfaction-based generator of topical Indonesian poetry. In *Proceedings of the Fifth International Conference on Computational Creativity*, 82–90. Ljubljana, Slovenia: Association for Computational Creativity.

[Salzberg et al. 1999] Salzberg, S. L.; Pertea, M.; Delcher, A. L.; Gardner, M. J.; and Tettelin, H. 1999. Interpolated Markov models for eukaryotic gene finding. *Genomics* 59(1):24–31.

[Simonton 1990] Simonton, D. K. 1990. Lexical choices and aesthetic success: A computer content analysis of 154 Shakespeare sonnets. *Computers and the Humanities* 24(4):251–264.

[Simonton 2011] Simonton, D. K. 2011. Creativity and discovery as blind variation: Campbell's (1960) BVSR model after the half-century mark. *Review of General Psychology* 15(2):158.

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

[Ward, Smith, and Finke 1999] Ward, T. B.; Smith, S. M.; and Finke, R. A. 1999. Creative cognition. In *Handbook of creativity*. Cambridge, UK: Cambridge University Press. 189–212.