Constrained creation of poetic forms during theme-driven exploration of a domain defined by an N-gram model

Pablo Gervás

Published online: 03 Feb 2016.

Submit your article to this journal ⬈

Article views: 134

View related articles ⬈

View Crossmark data ⬈

Taylor & Francis
Taylor & Francis Group

Constrained creation of poetic forms during theme-driven exploration of a domain defined by an N-gram model

Pablo Gervás

Instituto de Tecnología del Conocimiento, Universidad Complutense de Madrid, Madrid, Spain

ABSTRACT

Most poetry-generation systems apply opportunistic approaches where algorithmic procedures are applied to explore the conceptual space defined by a given knowledge resource in search of solutions that might be aesthetically valuable. Aesthetical value is assumed to arise from compliance to a given poetic form – such as rhyme or metrical regularity – or from evidence of semantic relations between the words in the resulting poems that can be interpreted as rhetorical tropes – such as similes, analogies, or metaphors. This approach tends to fix a priori the aesthetic parameters of the results, and imposes no constraints on the message to be conveyed. The present paper describes an attempt to initiate a shift in this balance, introducing means for constraining the output to certain topics and allowing a looser mechanism for constraining form. This goal arose as a result of the need to produce poems for a themed collection commissioned to be included in a book. The solution adopted explores an approach to creativity where the goals are not solely aesthetic and where the results may be surprising in their poetic form. An existing computer poet, originally developed to produce poems in a given form but with no specific constraints on their content, is put to the task of producing a set of poems with explicit restrictions on content, and allowing for an exploration of poetic form. Alternative generation methods are devised to overcome the difficulties, and the various insights arising from these new methods and the impact they have on the set of resulting poems are discussed in terms of their potential contribution to better poetry-generation systems.

1. Introduction

Computer generation of poetry is opportunistic in as much as it brings together mechanisms for the automatic production of text with procedures for identifying features that might have an aesthetic value. If one can generate a sufficiently large set of candidate drafts, and one can spot rhyme or metric regularity, systematic search will eventually produce poem-like structures. The simplicity of this approach explains both the proliferation

of computer poets and the distaste of their many detractors. In the last few years there has been a significant increase in the number of approaches to the task, and an extension to work in languages previously untried. However, if computer generation of poetry is ever to be taken seriously, it must start exploring approaches that are progressively less opportunistic and more goal-driven. To be valuable, computer generated poetry must certainly have aesthetic value, but, beyond the pioneering efforts that defined it as a field, it must also make progress towards outputs that also have a valuable meaning. This point was made very clearly by Manurung in his Ph.D. thesis (Manurung, 2003), but tends to be forgotten in later efforts.

The quality of good poetry emerges from a delicate balance between form and content. Content should not arise randomly as a result of the exploration of a conceptual space in search of aesthetic value. Yet existing poetry-generation systems tend to do just that. By its nature, the task of generating a poem, when addressed by either a computer or a human, has to satisfy constraints at two very different levels. One level concerns the sequence in which the words appear in the poem. For a draft to be acceptable there has to be some way in which the words in it appear to link to one another, to make sense as a linguistic message. This constraint is applicable to the whole poem but essentially it operates at a local level, based on how each word can be seen to follow on from the previous one. A different level concerns certain macro-structural features that may be desirable in a poem, such as being distributed over a number of lines of specific lengths in terms of syllables, or having rhyming words occur at the end of particular lines. This corresponds to the poem satisfying some form of poetic stanza.

The problem of poetry generation is in fact rather more complex, because these two levels of constraints are just formulations of the overall specification at the extremes of a continuum. In truth, the way in which the sequence of words builds up is also expected to satisfy constraints on form – usually based on the relative positions of stressed syllables within a line, sometimes expressed in terms of feet – and there must also be some sense to be made between the different parts of the poem at a linguistic level. This is why human quality poetry is a tall order that few computer programs can tackle to the satisfaction of their critics. However, two higher level characteristics of poetry can be exploited to simplify the problem from an engineering point of view. First, poetry can also exist in free form, where constraints on line length, stress patterns or rhyme may be waived in favour of a more expressive poem at a semantic level. Second, the concept of poetic licence allows poets to sometimes violate linguistic expectations in favour of a more pleasing poem in terms of form. Traditionally, these two characteristics are applied in opposition to one another: if free-form is chosen for a poem, it is usually so that its linguistic expression does not have to be forced in any way to express the poet's meaning; if poetic licence is applied, it is usually to fit the poet's meaning into a particular poetic form where conventional phrasings might not work. Computer generated poetry often operates at the confluence of these two approaches relying on one to avoid the need to achieve rigorous poetic form and on the other to avoid the need of conveying a specific message at a semantic level. As the full problem is so complex, it is acceptable to apply a certain degree of simplification so that progress can be made in spite of the difficulties. However, the original goal must be kept in mind, so that once acceptable solutions have been found for the simplified version of the problem, progress can be made towards it by enriching the initial problem statement.

The present paper describes such an attempt. An existing computer poet, originally developed to produce poems in a given form but with no specific constraints on their content, is put to the task of producing a set of poems with explicit restrictions on content, and allowing for an exploration of poetic form. The constraints being considered arise from the fact that the set of poems to be produced had been commissioned for inclusion in a book written by a Mexican author. The approach previously followed to poetry generation is shown to have limitations when the task is rephrased in this way. These limitations are analysed in terms of the current theoretical descriptions of computational creativity, and alternative generation methods are explored.

2. Previous work

The work presented in this paper brings together some of the existing theoretical accounts of computational creativity and a number of efforts for computer generation of poetry. Both of these separate topics are reviewed in the present section.

2.1. Computational creativity

Much of the work done on computational creativity over the past few years has been informed by Margaret Boden's seminal work describing creativity in terms of search over a conceptual space (Boden, 1990). Boden formulated the search of ideas in terms of search over a conceptual space. Such a conceptual space would be defined by a set of constructive rules. The strategies for traversing this conceptual space in search of ideas would also be encoded as a set of rules. This view of computational creativity was taken a step further in Wiggins (2006) by specifying formally the different elements involved (the universe of possible concepts, the rules that define a particular subset of that universe as a conceptual space, the rules for traversing that conceptual space, and a function for evaluating points in the conceptual space reached by these means).

In his pioneering work on the evaluation the creativity of computer programs, Ritchie (2007) outlined a set of empirical criteria to measure the creativity of the program in terms of its output. Ritchie's criteria are defined in terms of two observable properties of the results produced by the program: novelty (to what extent is the produced item dissimilar to existing examples of that genre) and quality (to what extent is the produced item a high-quality example of that genre). He also put forward the concept of *inspiring set*, the set of (usually highly valued) artefacts that the programmer is guided by when designing a creative program. Ritchie's criteria are phrased in terms of: what proportion of the results rates well according to each rating scheme, ratios between various subsets of the result (defined in terms of their ratings), and whether the elements in these sets were already present or not in the inspiring set.

This idea of the inspiring set was taken a step further in Gervás (2011), where the issue of how systems might take their prior output into account when evaluating the novelty of subsequent artefacts. This lead to the introduction of the concept of a *dynamic inspiring set*, one where system outputs are progressively updated into the inspiring set so they can inform later generative processes.

Colton and Wiggins (2012) introduced the term *curation coefficient* to identify the proportion of system results that an impartial observer of system output would be happy to

present to third parties. When estimated for a system addressing creative tasks it provides a reasonable measure of how much of the merit of presented system output can be attributed to the system itself and how much to the person actually selecting which particular outputs to present.

2.2. Computer generated poetry

Computer generation of poetry has traditionally addressed the constraints outlined in Section 1 in terms of two different strategies: one is to reuse large fragments of text already formatted into poem-like structures of lines (Das and Gambäck, 2014), and the other is to generate a stream of text by some procedure that ensures word-to-word continuity and then establish a distribution of the resulting text into lines by some additional procedure.

The reuse of text fragments already distributed into poetic lines was pioneered by Queneau (1961), Oulipo (1981) and it has more recently been used by Toivanen, Toivonen, Valitutti, and Gross (2012), Gonçalo Oliveira (2012), Colton, Goodwin, and Veale (2012), Veale (2013), Toivanen, Gross, and Toivonen (2014), Charnley, Colton, and Llano (2014), Rashel and Manurung (2014). In all these cases, either lines or larger poem fragments from existing poems are subjected to modifications – usually replacement of some of the words with new ones – to produce new poems. A refinement on this method the selected fragment is stripped down to a skeleton consisting only of the POS tags of each line, and words corresponding to the desired content are used to fill this skeleton in. This procedure is followed in Gervás (2000), Agirrezabal, Arrieta, Hulden, and Astigarraga (2013), Toivanen, Järvisalo, and Toivonen (2013).

Alternative procedures rely on building a stream of text from scratch, and resort to various techniques to ensure the continuity of the textual sequence. One early approach was to rely on linguistic grammars to drive the construction. This was the approach followed in Manurung (1999, 2003), where TAG grammars were employed. A more popular alternative is the use of n-grams to model the probability of certain words following on from others. This corresponds to reusing fragments of the corpus of size *n*, and combining them into larger fragments based on the probability of the resulting sequence. This is the main approach for ensuring text coherence used in Barbieri, Pachet, Roy, and Esposti (2012), Gervás (2013a), Gervás (2013b) and Das and Gambäck (2014). All these different computer poets rely on various additional methods for establishing constraints on the resulting poem drafts.

To ensure that resulting poems satisfy constraints on poem structure in terms of lines, systems that build a stream of text from scratch rely on either building each line separately (Das and Gambäck, 2014) or applying a separate procedure for distributing the resulting text into poetic lines (Gervás, 2013a,b).

2.3. The WASP system

The development described in this paper was carried out over an existing version of the WASP system (Gervás, 2013a,b).

Combining n-gram modelling and evolutionary approaches, the WASP poetry generator had been built using an evolutionary approach to model a poet's ability to iterate over

a draft applying successive modifications in search of a best fit, and the ability to measure metric forms. It operates as a set of families of automatic experts: one family of content generators or *babblers* – which generate a flow of text that is taken as a starting point by the poets –, one family of *poets* – which try to convert flows of text into poems in given strophic forms –, one family of *judges* – which evaluate different aspects that are considered important –, and one family of *revisers* – which apply modifications to the drafts they receive, each one oriented to correct a type of problem, or to modify the draft in a specific way. These families work in a coordinated manner like a cooperative society of readers/critics/editors/writers. All together they generate a population of drafts over which they all operate, modifying it and pruning it in an evolutionary manner over a number of generations of drafts, until a final version, the best valued effort of the lot, is chosen. In this version, the overall style of the resulting poems is strongly determined by the accumulated sources used to train the content generators, which are mostly n-gram based. Several versions have been developed, covering poetry generation from different inspirational sources as different sets of training corpora are used: from a collection of classic Spanish poems (Gervás, 2013a) and a collection of news paper articles mined from the online edition of a Spanish daily newspaper (Gervás, 2013b). Readers interested in a full description are referred to the relevant papers. However, two specific aspects of this implementation are relevant for the present paper. First, the various judges assign scores on specific parameters – on poem length, on verse length, on rhyme, on stress patterns of each line, on similarity to the sources, fitness against particular strophic forms... – and an overall score for each draft is obtained by combining all individual scores received by the draft. A specific judge is in charge of penalising instances of excessive similarity with the sources, which then get pushed down in the ranking and tend not to emerge as final solutions. Second, poets operate mainly by deciding on the introduction of line breaks over the text they receive as input.

3. A revised constructive procedure capable of undertaking a commission for a set of themed poems

The work reported in this paper arose in response to a request received by the author to provide a set of poems generated by the WASP poetry system to be included in a book chapter about computational creativity. The request explicitly indicated that these poems should never have been published anywhere else, to avoid possible problems with copyright. Additionally, the author decided that the poems should aim to achieve a certain thematic unity, somehow relating to the circumstances in which they were commissioned. Finally, the author wanted to include data on what percentage of all outputs generated for the system was considered valuable to fulfil the original commission.

These conditions posed a challenge to the existing implementation of the WASP system. First, because the system as it stood had no means for driving the resulting poems towards particular themes. Second, because the procedures already in place for ensuring originality were inefficient. Third, because prior versions of the system had relied on significant post-processing to select outputs to be presented: only a very small subset of actual system output was worthy of presentation to a wider audience.

The final set of poems was achieved by a recombination of some of the existing modules with new modules specifically designed for the occasion, and by a new procedure for

generating poems that abandoned the original generate and test approach underlying the evolutionary version of the system for a more informed generative approach that applied backtracking in search of solutions that better fulfilled the driving constraints.

The new constructive procedure is also based on n-gram-based generation of text, but relies on a language model trained over a corpus specifically constructed to match the goals of the commission (Section 3.1).

A two-stage procedure for building poems is applied. First, a number of sentence-based poem fragments is constructed following an adaptation of the original generative procedure of the WASP system (Section 3.2). Then, a recombination procedure is applied to combine selections of these poem fragments into larger stanza-like drafts (Section 3.3).

3.1. Stage 0: Compiling a corpus to train the language model for the themed commission

As the book for which the poems were commissioned was to be published in Mexico, it was decided that the poems should have a Mexican theme. As the babbler modules rely on an n-gram model of language to produce sequences of text that are word to word coherent, the overall style of the resulting poems is strongly determined by the accumulated sources used to train the content generators. For this initiative, a corpus of training texts was constructed by combining an anthology of poems by Mexican poets compiled from the Internet, and a set of news articles mined from the web pages of an online Mexican daily newspaper. The anthology of poems by Mexican poets included works from 23 authors. The set of poems had an overall length of 35,183 words. The set of news articles had 340 new items and 112,519 words.

Earlier attempts to generate based on the simpler model trained only over the set of news items resulted in a candidate texts that were very difficult to adjust to any given poetic form. This related to the fact that the sequences of words contemplated in the n-gram model resulting from news items only did not include enough combinations with a potential for poetic form. When the training set was expanded with an additional set of poetic texts, the resulting set of candidate texts showed a greater potential for composition into poetic forms.

This observation corroborates the intuition that the set of training texts used to train the n-gram model imposes a certain overall style on the texts that can be produced. But it also raises the question of whether the desired poetic form is obtained at the price of replicating fragments of the poems being used as part of the inspiring set. This issue is addressed below.

3.2. Stage 1: Construction of sentence-based poem fragments

The generative procedure for this module of the system involved successive application of:

- a constructive procedure that produced sentences,
- a composition procedure that aimed to distribute those sentences over lines of equal length in syllables to construct poem drafts, and

- a set of judges that filtered out poem drafts that violated any of the requirements imposed.

Each of these stage is explained in more detail in a separate section below.

3.2.1. Constructive procedure

At least two improvements to the original generative procedure of the WASP system were required to fulfil the goals we had set out to achieve. One was to improve the fitness functions overall so that only results of a higher quality survived the evaluation stage. Another was to somehow improve the construction procedure itself so that better quality results were produced. The evolutionary paradigm of the original approach required mostly random procedures for generation and revision, with quality to be achieved by means of evolutionary operators combined with selection in terms of the fitness function. But this approach clashes with the fact that the conceptual space that we want to explore is constrained to the set of texts that can be derived from the n-gram model under consideration. For the evolutionary operators to guarantee that mutation and cross over produce results that are still within the desired conceptual space, they would have to be restricted to operations that take into account the n-gram model during mutation and/or cross over.

The option of refining the revisers by enriching them with knowledge so that the changes they introduced were more informed was seen as impractical, and it was preferred to overhaul completely the generation procedure so as to take advantage of the available information to only generate valuable results in the first place.

The revised version of the construction procedure expanded the initial solution for babblers, which was based on extending a candidate sequence of words with further words that have a non-zero probability of appearing after the last word of the sequence, according to the n-gram model. In both versions, at each choice point, the system is faced with a number of possible continuations. In the earlier version, this choice was taken randomly. In the new version, the choice is made taking into account additional criteria, covering the following issues: relation to theme, plausibility of sentence ending, control over repetition of sentences already generated, and restriction to overall length of sentences.

The first criterion to consider involves the initial constraints on theme, giving preference to options related to the desired theme.

The second criterion is designed to rule out cases where a draft is ended at a point where the word sequence under consideration does not allow the ending of the sentence.

The third criterion aims to avoid having the system repeat itself. A model of short-term memory for sentences has been added, so that continuations of sentence drafts that replicate sentences constructed recently are avoided.

The final criterion ensures that text candidates are restricted to single sentences, and the overall length is restricted by introducing a check on the accumulated length of the word sequence that starts giving priority to continuations that close off the sentence after a given threshold length has been achieved.

It must be noted that in applying these criteria, the system considers all possible continuations of non-zero probability. This is to ensure that the search space of all acceptable sentences is explored, rather than just the most probable ones.

3.2.2. Composition procedure

The procedure for composing candidate texts into valid poetic forms is revised in the following way. For any given candidate text the poetic composition module:

- finds the set of line lengths that have a potential to give an exact break down of the total number of syllables in the text,
- composes a number of candidate draft poems based on the input text, each one distributing the text into lines of the corresponding length as worked out above,
- evaluates the resulting set of poem drafts , and
- returns only those that are positively evaluated in terms of the judges for metric form.

3.2.3. Filtering procedure

The set of judges is revised so that drafts in any one of the following situations are ruled out directly:

- candidate drafts with line lengths beyond 14,
- candidates drafts that have lines of different lengths.

The fitness function for originality was addressed by developing a specific judge module that held the complete set of texts in the training set as a master file. Every line appearing in a candidate draft was searched for in the master file, and the candidate draft was rejected if the particular sequence of words in any of its lines appeared as a continuous unit anywhere in the master file. This ensured that only lines that combined elements from different parts of the training set in innovative ways were considered by the system.

The set of requirements imposed by the final set of judges employed for the actual generation of the poems for the commission can be summarised as

- no lines of more than 14 syllables,
- no drafts with lines of unequal length, and
- no draft that includes lines already appearing in the training set.

3.2.4. Overview of intermediate results of stage 1

The described adaptations result in an exploratory software that takes a long time to run – as the constructive procedure explores exhaustively the portions of the conceptual space established by the given n-gram model that include words from the desired theme – and produces a much smaller set of candidate drafts than earlier versions. These candidate drafts are of high quality in terms of poetic form – they correspond to stanzas of lines of the same length in syllables – but are surprisingly short in length – they very rarely exceed two lines. This restriction on length is a result of the interplay between the configuration that limits texts to single sentences and the restriction that the system start trying to close sentence as soon as a minimally valid length has been reached.

This set of results is not in itself a convincing set of poems with which to satisfy the received commission. But it constitutes a treasure trove of valuable material generated by the system: it is by construction innovative – in terms of p-creativity as described by Boden, given that the originality judges check each line against the master file built from the training corpus and rule out any replications – and it is remarkable in its poetic

form – as guaranteed by the remaining judges. It is a small set, but large enough to allow a further step of recombination of these poem snippets with one another.

3.3. Stage 2: Recombination of sentence-based poem fragments into stanza-like drafts

The construction procedure was therefore extended with a further stage that considered these poem drafts as possible ingredients to combine into larger poems. The heuristics considered to drive this recombination process were as follows:

- the set of poem snippets was classified into groups according to the length of their lines in syllables,
- poem snippets of the same length of line were further grouped together into sets related by shared rhymes , and
- larger poem drafts were built by combining together the sets of snippets of the same line length that had shared rhymes.

The initial set of small poem drafts was produced in six separate runs with the same configuration, designed to carry out 1000 attempts to build poem drafts fulfilling the constraints as described above. The data on number of valid poem drafts found in each of these runs are presented in Table 1. There seems to be no consistency on the number of valid drafts resulting from individual runs. This results from the interaction between the random ingredient in the construction process and the complexity of the conceptual space that is being searched.[1]

The average rate of success over this limited set of data – excluding the data for aborted runs on the grounds that no record is available of the number of attempts they had carried out before being stopped – is 13.5%. Given the complexity of the conceptual space that is being searched, this rate is considered very acceptable.

The total number of snippets obtained in this way that was used as input for the procedure for composing larger poem drafts was 469.

The procedure for recombining the generated poem snippets into larger poem drafts produced 42 poem drafts, as described in Table 2. Overall these poems have used 18 different rhymes, irregularly spread over the set of resulting poems. The numbers provided for the complete set of poems do not correspond to the addition of the specific values for different line lengths because poem lengths and rhyme schemes are sometimes repeated for different line lengths.

Table 1. Rates of success in the runs for collecting an initial set of poem snippets.

Run #	Valid drafts found
1	149
2	46
3	106
4	150
5	8
6	10

Table 2. Description in terms of line lengths of the set of poem drafts obtained by recombination of snippets.

Line lengths	Poems	Poem lengths	Rhyme schemes
6	1	1	1
7	9	6	7
8	6	3	6
9	10	7	10
10	14	8	14
12	2	1	–
All	42	11	31

Table 3. Description in terms of the shorter stanzas in the set of poem drafts obtained by recombination of snippets.

Stanza size	4	5	6	7	8	9
Rhyme schemes	ABBA	BACDA	BAAAAB	CBADABA	BACADAAE	ABABCBBDB
	ABAB	ABCDA	ABABAB	BACDAAE		ABCDEAFBA
	AAAA		ABCABC			ABACDEAFA
	BACA		ABABCA			BACADAEAF
	ABAC		ABABAC			ABACDAAEF
	ABCA		BAACAA			
			ABCADA			
			ABCAAD			
			ABACAD			

Table 4. Description of the longer stanzas in the set of poem drafts obtained by recombination of snippets.

Stanza size	Rhyme schemes
10	ABABCAABDA
14	BAABACADAEFBAA
	ACABABADAEAFCA
15	EACAFGBADADAHCA
20	AEACABABFAGAADADACAC
21	BACBAADAAAEAFAGHAAIJA

The poems that resulted from this process were of different size, and for each particular poem size a rhyme schema results from the way in which snippets sharing rhyming lines have been combined. The analysis of the resulting set in terms of these emerging stanzas and rhyme schemes is presented in Tables 3 and 4.

4. Results

Of the 42 poems generated, 13 poems were deemed to be unusable as a result of problems in the generation process. The type of problems that were identified included issues of incorrect scanning of line lengths due to the appearance of punctuation signs not covered by the parsing procedures (2), undesirable repetition of subsets of lines (5), occurrence of unknown words (4), inclusion of unacceptable rude words (2).

The issue with incorrect scanning of line lengths has now been corrected.

Repetition of fragments of poems of more than one line is discouraged. The ones appearing in the result set have been tracked down to a small bug in the recombination process that should be easy to fix.

Some of the unknown words appear because the corpus of news items is mined directly from the web and the pre-processing procedures applied to clean up the html code sometimes miss non-words that end up in the training set. Improvements on the clean up procedure already under way should avoid this problem in the future.

Another source of problematic words is the use of foreign languages proper names, also frequent in news items. These words are acceptable in terms of their semantics contribution but their spelling confuses the metric analysis module of the system – specifically tailored for the phonetics of Spanish – which computes an incorrect number of syllables for them. This in its turn affects the composition processes that convert the resulting text into poetic form.

Rude words seem to have been used in some of the news items in the corpus, or possibly in some of the poems. But they are not considered desirable for the commissioned set of poems.

Of the remaining 29 poems, 7 were selected to be included in the book chapter that gave rise to the commission. This selection was based on general quality, but also on how well the selected poems fitted the desired theme. The 22 poems that were not selected show acceptable quality, and they were excluded from the selection for one of the following reasons:

- they shared some lines with the poems already selected,
- their relation to the desired theme was not clear,
- they included mentions of entities too specific to Mexican current news to be easily identified by a general public,
- they included proper names of individuals featuring in the Mexican news, and
- they were overlong.

Example results of the poems produced in this way are presented in Table 5. These examples correspond to a second stage of selection out of the 22 poems that had not been chosen for inclusion in the set of poems commissioned for the book chapter.

The poems numbered 1, 2, and correspond to four-line poems of different number of syllables per line (7, 7, 8 and 9, respectively), and showing different rhyme schemes (BACA, ABCA, ABAC). Together they illustrate the ability of the system to find the most metrically appropriate form for presenting a given text, using different lengths of line in syllables as required. They also illustrate the ability of the system to operate with different rhyme schemes to make the most of a given text.

Poem number 4 is made of 4 *eneasílabos* of 9 syllable lines. Lines 2 and 3 share an asonant rhyme in *i-a*. The restriction on early closure of sentences has produced here a certain *staccato* feeling that is in line with the topic being addressed. Serendipity has led to a marked contrast between "military" and "delight", followed up with a surprisingly appropriate "incomprehensible". In spite of the choppy phrasing, as "girl friend" and "delight" agree in gender in Spanish, there is an implicit thread to the first two lines that is quite evocative. The third line mentions the female proper name "Agueda", rounding up this impression. This is again serendipitous. But it poses the question of whether similar criteria might not

Table 5. Example set of poems generated by the system (excluding those used for the original commission and those rejected according to the criteria presented in Section 4), with an approximate English translation.

Poem 1: 4 lines of 7 syllables with rhyme scheme BACA

Toda era una ave larga	Everything was a long bird
que cuando se conforman.	that when they conform.
Admitió que se tienen	He admitted that they have
registradas personas.	registered persons.

Poem 2: 4 lines of 7 syllables with rhyme scheme ABCA

Muestra también. Esta	Shows as well. This
noche adonde yo soy.	night were I am.
Subraya que para ellos	Underlines that for them
ya no salgas. Estrellas.	come out no more. Stars.

Poem 3: 4 lines of 8 syllables with rhyme scheme ABAC

Aspecto que se encontraban	Aspect that they were finding
ejemplares. Nuevamente.	exemplars. Again.
Señalaron que no haya	They pointed out that there should not be
más daños como los niños.	more harm like the children.

Poem 4: 4 lines of 9 syllables with rhyme scheme ABAC

Zona militar. Qué delicia	Military zone. What a delight
delgada incomprensible. Amiga.	thin incomprehensible. Girl friend.
Agueda era luto pupilas	Agueda was mourning green
verdes. Sobrepasa. Guerrero.	pupils. Overshoots. Warrior.

Poem 5: 6 lines of 9 syllables with rhyme scheme ABABAB.

Juegan el largo recorrido	They play the long tour
desde su muerte ya no salgas.	from his death come out no more.
Séptimo. Cordero tranquilo	Seventh. Peaceful lamb
cordero que paces tu grama.	lamb that grazes its grass.
Silencios. Cordero tranquilo	Silences. Peaceful lamb
cordero que paces tu grama.	lamb that grazes its grass.

Poem 6: 8 lines of 10 syllables with rhyme scheme BACADAAE.

Engalanados por los derechos	Garlanded by the rights
del niño indígena. Apago soles.	of the indigenous child. I switch off suns.
Concluido el objetivo que exista	Having achieved the goal that it exists
todo el mes para que ya sin nombre.	the whole month so that now nameless.
Dichosa puerta que nos transforman.	Happy gate that they transform for us.
Solidaridad vocación. Hombres.	Solidarity vocation. Men.
Acción nacional tiene un enorme	National action has an enormous
pez que se ilumina. Guatemala.	fish that lights up. Guatemala.

Poem 7: 15 lines of 9 syllables with rhyme scheme CAABADAEAFGBAA.

Tus ojos. Vinos tintos	Your eyes. Red wines
blancos rosados. Nardo.	whites rosés. Tube rose
Amo tus ríos claros.	I love your clear rivers.
Tal vez esta medalla.	Maybe this medal.
Antes que este hachazo	Before this axblow
nos sacude. Imagínate.	shakes us. Imagine.
Séptimo. Pinté el tallo	Seventh. I painted the stem
luego el cáliz después.	then the calyx afterwards.
Ganar. Solicitamos.	Winning. We request.
Sólo soy un prisionero.	I am only a prisoner.
Admitió que se estima	He admitted that it is estimated
que mil michoacanas	that a thousand Michoacans
acudan. Queretanos.	turn up. Queretans.
Valor. Acompañado	Valour. Accompanied
por Margarita Flores.	by Margarita Flores.

be used to derive selection heuristics so that future versions of the system can attempt to achieve similar effects. The final word "warrior" is ambiguous, and may originally have been intended as a reference to the Mexican state of Guerrero, but also links up with the military theme.

Poem number 5 is composed of 6 *eneasílabos* of 9 syllable lines. Lines 1, 3 and 5 rhyme together, and so do lines 2, 4 and 6. The rhyming is poor because it basically involves some of the line endings being repeated twice. However, this arises from a parallelism trope - same linguistic structure used repeatedly with slight variations of content – and this makes the repeated rhyme somewhat more acceptable. The repetition is serendipitous and arises from the fact that particular sequences of words that match well a given poetic form tend to be reused to fill in certain stanzas ("Cordero tranquilo // cordero que paces tu grama."), relying on different fragments of similar length to cover the initial first few syllables ("Séptimo.", Silencios."). Remember the constituent snippets were originally built separately, and they are only combined by application of the described composition heuristics. The apparent rhetorical effect is a consequence of the interaction between the limitation in the constructive procedure for poem snippets and the composition heuristics. Having noticed this interaction, we hope to include it as a system feature in future releases. In this particular case, the sequence in which the different fragments appear also achieves a significant effect, with the neighbouring mention of "death" and "lamb" evoking a certain hint of Christian symbolism. The effect of the early closing policy for sentences is also apparent in this poem.

Poem number 6 is composed of 8 *decasílabos* of 10 syllable lines. Lines 2, 4, 6 and 7 rhyme together. It presents interesting features that arise from the fact that sentences in the news items corpus are not generally well suited for partition over several valid metric lines, which lead to them being cut off abruptly at points where the closure makes syntactic sense. The texts in the poetic part of the corpus perform better in this sense, possibly as a result of being composed with metric form in mind. Drafts where the system alternates fragments from the two different parts of the corpus tend to achieve greater sentence lengths, as well as interesting contrasts between day to day pragmatic topics arising from news items and grander and more abstract topics obtained from the poems in the corpus. The Mexican theme is hinted at by the mention of the indigenous child.

Poem number 7 is included as an example of a longer poem. It has 15 *heptasílabos* or 7 syllable lines. Lines 2, 3, 5, 7, 9, 13 and 14 share asonant rhyme in *a–o* and lines 4 and 12 share asonant rhyme in *a–a*. This results in a rhyme scheme of the form CAABADAEAFG-BAA. The Mexican theme is apparent in the mentions of citizens of two different Mexican states ("michoacanas", women from Michoacán; and "Queretanos", men from the town of Querétaro).

5. Discussion

Several aspects of the described work need to be discussed: the relative merit of the construction method employed, the justification of the changes introduced with respect to the original approach, and valuable insights derived from the work presented here that might be relevant to computational creativity at large.

5.1. Advantages and disadvantages of the construction method

Over the complete run, 29 out of 42 poems were considered acceptable according to the criteria described in Section 4. Of those 29, 13 have been submitted for publication in different media – either the book chapter for which the poems were commissioned or the present

paper –, with the reasoning for their selection in each case also argued in that section. The remaining 16 poems are less impressive but acceptable overall – they are not included here for lack of space –, although they do have the disadvantage of sharing some lines with the preferred poems. This, however, should not be considered as a demerit of the poems themselves. Instead, it should be thought of as an issue of incompatibility between possible system outputs in terms of originality. Once one particular line has been included in a poem submitted to the public, the system should refrain from including it in further output. This issue had already been described in Gervás (2011), and more attention should be paid to it in poetry generators in the future.

These numbers lead to a percentage of system output considered valuable from the described run of around 69%. Unfortunately, no record has been kept of equivalent values for earlier versions of the system. This is in part because the earlier versions produced much larger numbers of candidate poems, and the set of candidates was never explored exhaustively. Instead it was sampled at random and poem instances that seemed valuable to the developer were picked out for public presentation. This reduces the objective significance of comparisons between the data on both approaches. However, it remains worthy of note that a shift in the approach to selection for presentation has taken place.

An interesting feature of the system described in this paper is that instead of establishing as configuration parameters values for features such as number of lines, number of syllables per line, or rhyme scheme to use, it relies on an exploratory procedure that allows the system to find optimal values for these features depending on the text that it has to convey. This leads to the variety of line and poem lengths, and the broad range of rhyme schemes that appear in the result set.

This variation in the range of rhyme schemes might be presented as an argument in favour of the perceived creativity of the system. The exploratory procedure in place relies on a fitness function that assigns higher value to poems that exhibit rhyming lines, but it does not prescribe any particular patterns for the rhymes. This results in output that satisfies rhyme schemes not traditionally used by human poets. This could be interpreted as a shortcoming, but it can also be considered as a creative feature.

The reliance on a corpus of training texts to produce candidate texts to compose into poetry introduces a number of dependencies between the particular training set chosen and the range of output text that can be generated. In the examples above this has been shown to lead to poems satisfying certain thematic constraints, not necessarily arising from explicit theme related constraints but simply as a result of having constrained the corpus to text somehow related to the theme. The issue of explicit constraining on theme needs to be explored further.

The influence of the training corpus has also been shown to affect the plasticity of the resulting texts when trying to compose them into poetic metrical forms. Certain styles of prose, such as that used in news items, are less conducive to composing into metrically acceptable forms than those custom-composed for such a form of expression. This should be taken into account when building training corpora for this type of system. On the other hand, the combination of corpus elements coming from different domains can lead to interesting contrasts that may result in a perception of originality in the final results.

One interesting point is the role of punctuation. As a result of the way the n-gram models are constructed, most punctuation sign are stripped away from the texts before training. Question and exclamation marks are left in because they impact the syntax of the sentences they appear in. The output candidate texts are therefore generally devoid of punctuation. This introduces a degree of freedom that provides some leeway for human readers to find possible valid interpretation of the resulting poems. Readers should consider the possibility of revising the poems to consider whether simple punctuation, like the insertion of commas or semi-colons at certain points might improve them. It is after all, a task that editors of poetry sometimes do take out of the hands of their poets, even when they are human. In any case, having noticed the possible significance of this issue, the development of a system module to address such a refinement task is being considered as future work.

The striking effect induced by poem 5 – see Table 5 – with neighbouring mentions of "death" and "lamb" evoking hints of Christian symbolism – suggests that a large amount of the attribution of value to a particular poem may be the result of interpretative work by the reader rather above and beyond the intentions of the writer. This relates to the idea of the "death of the author" as argued by Postmodernist thinkers (Barthes, 1978). The need to extend current models of computational generation of literary texts to include explicitly this type of process has been argued elsewhere (Gervás and León, 2014), and should be considered as a valuable addition for further work.

A final question to be considered is that of the originality of the output set in contrast to the inspiring set, here understood to correspond to the training corpus of texts. This question features prominently in Ritchie's set of criteria for evaluating the output of creative systems (Ritchie, 2007). The system presented in this paper includes by construction a filter on candidate poem drafts that rejects them if they include a line that can be found as a continuous sequence of words anywhere within the training set. This should ensure that no line in any of the resulting poems correspond to lines in the poems in the training set, and it should also reduce significantly the chance that sentences in the training corpus are replicated verbatim.

5.2. Comparison with earlier approach

The original WASP evolutionary system was designed to produce an initial large population of drafts – based on its n-gram-based babbler modules –, to compose these into poem drafts by inserting line breaks at appropriate places – relying on its poet modules –, and to select as output a quality subset from those candidate drafts by applying the fitness functions implemented in its judge modules. This procedure was effective because it allowed the system to zoom in towards the regions of the overall conceptual space – as defined by the n-gram model of language being used – that held potentially valuable text fragments from the point of view of poetic form – as defined by the fitness functions. This procedure was reasonable when the only constraint on the result was that it satisfy a certain poetic form. Specific poet modules and fitness functions would be designed for the particular poetic form, say, for a *cuarteto*, and the system would explore all the possible poems of this form arising from the given n-gram model. This approach had two disadvantages for the present initiative: one related to form and one related to theme.

The existing solution was devised to drive the system towards poems of a particular type. When giving priority to theme in the new solution, a certain flexibility in form could be introduced, allowing for poems with different poetic forms as long as they were consistent with the theme. To achieve this in terms of an evolutionary approach required the development of a confusing set of composition modules – capable of generating drafts in several poetic forms – and complex fitness functions – allowing for different fitness according to which particular poetic form was being considered. This lead to the consideration of alternative implementations.

The existing solution also had no obvious way of constraining results to particular themes. The word content of the results is constrained by the n-gram model used, but an n-gram model small enough to ensure that particular themes are present in the result would be too small to allow sufficient word recombinations to achieve valuable poetic forms. Additional elements could be added to the fitness function to rule out candidate drafts diverging from the desired themes, but this solution clashed with the decision above to consider alternative implementations.

A first attempt was carried out to simply redeploy the existing WASP modules – babblers, poets and judges – with the new purpose in mind. Under the new circumstances, judgements on candidate drafts could become more radical: if drafts were not related to the desired theme, they could be ruled outright. This had another consequence on the overall design: the reviser modules, which allowed exploration of the conceptual space by replacing certain words with others at random were seen to have little positive effect. Given the accumulated set of constraints on the results, random changes had a high probability of reducing fitness rather than improving it.

A formative evaluation was carried out over the existing prototype, configured so that a very large population of drafts was built, composed into a number of possible poetic forms, and evaluated using judges that combined fitness functions for theme, the various poetic forms considered, and originality. The revision modules were switched off for this test.

Fitness functions for theme relied on a set of input words to characterise the desired theme, penalising the drafts that did not include any of them, and reinforcing the drafts that did.

Fitness functions for poetic forms were already available as judge modules, and a simple combination of judges for different poetic forms was employed.

The fitness function for originality applied in the solution reported in this paper was originally developed for this formative experiment.

This approach generated a very large set of results but with very low average quality. This might have been acceptable if the set of results was mined for valuable drafts, but this would imply a very low curation coefficient for the final set.

Nevertheless, the modularity of the original system has been retained to a certain extent. The module in charge of the construction procedure is an instance of the babbler modules of the original WASP system, the module in charge of the composition procedure is an instance of the poet modules of WASP, and the filtering procedure is implemented as a combination of judge modules. As a result, the system is open to addition of new implementations of these modules in place of existing ones. The original procedure, with its evolutionary approach, was well suited for parallel execution of more than one of these modules. The revised procedure is less so, though alternative implementations for Stage 1 –

as described in Section 3– could be used together to produce a broader set of intermediate results to be considered by Stage 2.

5.3. Valuable insight for computational creativity at large

Although many of the points outlined above deal with features that are specific to poetry, some of them can clearly be considered as valuable insights for computational creativity beyond poetry generation.

First, the idea that creative systems should evolve towards versions where the role of a human observer selecting a subset of system outputs as valid for publication is reduced to a minimum. This point had already been made by Colton and Wiggins (2012) with the introduction of the curation coefficient. However, the research described in this paper has shown that the definition they provide is only applicable as an abstract concept requiring an impartial observer. The distinct concept of the developer of a system carrying out a selection himself may need to be considered as a different definition, as such a developer would hardly count as impartial. In general terms, it is important that research efforts on computational creativity that involve the development of systems that generate output intended to be considered creative make a point of always reporting the percentage of their results that the developer is considering as valid. This would not necessarily correspond to a curation coefficient, as the developer could hardly be considered impartial, but it would be indicative of the role that the developer is expected to play in the preparation of system results for public presentation.

Second, the need to consider not only originality of each member of the output set with respect to the inspiring set but also with respect to other elements in the output set that may have been produced in parallel during the same run. The need to consider originality not only with respect to the inspiring set but also with respect to prior results of the system itself was already defended in Gervás (2011) via the introduction of the concept of a dynamic inspiring set. In a creative system with a dynamic inspiring set, each output of the system is added to the inspiring set as soon as it becomes available, and it thereby can be considered by the generation procedure both as inspiration and as an already known instance to be avoided in subsequent runs. This addressed the needs of considering the outputs of prior runs of the system when establishing the originality of a given run. The insights arising from the work presented in the current paper suggest that the issue should also be considered as part of the generative procedure itself, whenever a single run may produce more than one candidate artefact.

Third, the observation that, once the desired target is sufficiently specified, the introduction of randomness in the constructive procedure can have a negative impact. When generative procedures are defined as a search for aesthetical value over a conceptual space of possible artefacts, randomness may be a valid mechanism for driving the search. Whereas systematic exploration is likely to produce a sequence of candidate artefacts that differ very little from one to the next, randomness has the potential of producing at each subsequent call a candidate artefact that is radically different from the previous one. When aiming for an impression of creativity, this approach will obfuscate any tendency of the system to produce in different calls candidate drafts that resemble one another. No matter how convenient this might be in terms of development effort, solutions based on explicit consideration in the

construction procedure of the inadvisability of repetition should be preferred – as outline in the point on originality above.

Fourth, that tightening the constraints on the desired target is likely to lead to increases in the time taken to produce results, and to decreases in the amount of results produced. However, the results obtained in this way are more likely to be of high quality. Fast random search over a large space of possibilities driven by fitness functions that are quick to compute may lead to surprising results in little time. Computational creativity research should endeavour to move beyond this low hanging fruit to start addressing the complexities that lie beyond. On this challenging path, a very important aspect to consider is the effect of constraints on the output to fit a particular purpose. This has for a long time been considered a major requirement for creativity in the field of design, but is traditionally considered less important for more artistic endeavours. Recent approaches in the fielfring constraints in various forms (Barbieri et al., 2012; Toivanen et al., 2013). The present paper constitutes an initial attempt to address this point with respect to the general theme of the poems. This point can be related to the ongoing debate concerning the role of constraints in creative processes. This debate is too important to be addressed in detail in the present paper but interested readers can find a good overview in Rosso (2011).

6. Conclusions

The evolutionary solutions attempted in the past for poetry generation in the WASP system worked very well for the unconstrained exploration of broad conceptual spaces, where all parts of the space from a thematic point of view were equally valid as solutions, and constraints could be specified only in terms of metrical form. When constraints on theme are taken into consideration, it pays to relax the constraints on form, so that the system may look for the optimal poetical form covering a given theme. This has lead to the development of an exploratory procedure that sets its own values at run time for features such as poem length, line length, and rhyme scheme.

The refinement of the procedure for generating sentences to certain types of candidate – sentences of acceptable length and that can be understood as acceptably closed – had the consequence of restricting the possible outputs of the initial poem composition procedure to very short poem drafts. To compensate, a second stage of poem draft recombination has been added that builds larger poems from the set of initial candidate drafts. This recombination procedure is based on line length and shared rhymes, which leads to a result set that emulates reasonably well the composition of poems in terms of stanzas shaped together by rhyme.

Because this procedure for poem construction allows a certain freedom in terms of the shape of stanzas, the resulting set of output poems exhibits a number of different stanza-like structures, with different line lengths and rhyme patterns. This can be interpreted as an instance of creativity at the level of form. As such, this behaviour is innovative with respect to prior poetry-generation systems that either fixed the distinctive parameters of their output form or aimed for poems altogether free from formal constraints of any kind.

The ratio of acceptable system outputs over total system outputs is reported, and it is argued that such feedback on a generative system is informative and significant.

The analysis of system outputs has lead to the identification of a number of positive features that have been included by serendipity, but which hold a very high potential for

inclusion in future releases of the described system as quality-enhancing improvements. To handle these features might require an elaboration of the construction procedure as an interaction between a number of cooperating experts, in the way described in Misztal and Indurkhya (2014).

Note

1. For the sake of completeness, it must be noted that runs 2, 3 and 6 had to be aborted without finishing for practical reasons unrelated to system operation.

Disclosure statement

No potential conflict of interest was reported by the author.

Funding

ORCID

Pablo Gervás http://orcid.org/0000-0003-4906-9837

References

Agirrezabal M., Arrieta B., Hulden M., & Astigarraga A. (2013). *POS-tag based poetry generation with Wordnet*. In Workshop on natural language generation (ACL 2013).

Barbieri G., Pachet F., Roy P., & Esposti M. D. (2012). Markov constraints for generating lyrics with style. In Raedt L. D., Bessière C., Dubois D., Doherty P., Frasconi P., Heintz F., & Lucas P. J. F. editors. *ECAI 2012 – 20th European conference on artificial intelligence. Including prestigious applications of artificial intelligence (PAIS-2012) system demonstrations track, Montpellier, France, August 27–31, 2012, Frontiers in Artificial Intelligence and Applications*, Vol. 242, IOS Press. (pp. 115–120).

Barthes R. (1978). Death of the author. In *Image-music-text* (pp. 142–148).

Boden M. (1990). *Creative mind: Myths and mechanisms*. London: Weidenfeld & Nicholson.

Charnley J., Colton S., & Llano M. T. (2014). *The flowr framework: automated flowchart construction, optimisation and alteration for creative systems*. Paper presented at the 5th international conference on computational creativity (ICCC 2014). Ljubljana, Slovenia.

Colton S., Goodwin J., & Veale T. (2012). *Full-FACE poetry generation*. Proceedings of the international conference on computational creativity 2012, Dublin, Ireland. (pp. 95–102).

Colton S., & Wiggins G. A. (2012). Computational creativity: The final frontier? In Raedt L. D., Bessière C., Dubois D., Doherty P., Frasconi P., Heintz F., & Lucas P. J. F. editors. *ECAI 2012 – 20th European conference on artificial intelligence. Including prestigious applications of artificial intelligence (PAIS-2012) system demonstrations track, Montpellier, France, August 27–31, 2012, Frontiers in Artificial Intelligence and Applications*, Vol. 242, IOS Press. (pp. 21–26).

Das A., & Gambäck B. (2014). *Poetic machine: Computational creativity for automatic poetry generation in bengali*. Paper presented at the 5th international conference on computational creativity, ICCC 2014, Ljubljana, Slovenia.

Gervás P. (2000). *WASP: Evaluation of different strategies for the automatic generation of spanish verse*. Proceedings of the AISB-00 symposium on creative & cultural aspects of AI, Birmingham, UK. (pp. 93–100).

Gervás P. (2011). *Dynamic inspiring sets for sustained novelty in poetry generation*. Second international conference on computational creativity, México City, México.

Gervás P. (2013a). *Computational modelling of poetry generation*. Proceedings of the AISB'13 symposium on artificial intelligence and poetry, Exeter, UK.

Gervás P. (2013b). *Evolutionary elaboration of daily news as a poetic stanza*. Proceedings of the IX Congreso Español de Metaheurísticas, Algoritmos Evolutivos y Bioinspirados - MAEB 2013.

Gervás P., & León C. (2014). *Reading and writing as a creative cycle: The need for a computational model*. Paper presented at the 5th international conference on computational creativity, ICCC 2014, Ljubljana, Slovenia.

Gonçalo Oliveira H. (2012). *PoeTryMe: A versatile platform for poetry generation*. Proceedings of the ECAI 2012 workshop on computational creativity, concept invention, and general intelligence, C3GI 2012, Montpellier, France.

Manurung H. M. (1999). *Chart generation of rhythm-patterned text*. Proceedings of the first international workshop on literature in cognition and computers, Tokyo, Japan.

Manurung H. M. (2003). *An evolutionary algorithm approach to poetry generation* (PhD thesis). University of Edinburgh, Edinburgh, UK.

Misztal J., & Indurkhya B. (2014). Poetry generation system with an emotional personality. In *5th international conference on computational creativity, ICCC 2014*, Ljubljana, Slovenia.

Oulipo (1981). *Atlas de littérature potentielle*, Collection Idées, Vol. 1, Gallimard, Paris, France.

Queneau R. (1961). *100.000.000.000.000 de poèmes*, Gallimard Series, Schoenhof's Foreign Books Incorporated, Paris, France.

Rashel F., & Manurung R. (2014). *Pemuisi: A constraint satisfaction-based generator of topical indonesian poetry*. Paper presented at the 5th international conference on computational creativity, ICCC 2014, Ljubljana, Slovenia.

Ritchie G. (2007). Some empirical criteria for attributing creativity to a computer program. *Minds & Machines*, 17, 67–99.

Rosso B. D. (2011). *Creativity and constraints: Exploring the role of constraints in the creative processes of research and development teams* (PhD thesis). University of Michigan, Michigan, USA.

Toivanen J. M., Gross O., & Toivonen H. (2014). *The officer is taller than you, who race yourself! Using document specific word associations in poetry generation*. Paper presented at the 5th international conference on computational creativity, ICCC 2014, Ljubljana, Slovenia.

Toivanen J. M., Järvisalo M., & Toivonen H. (2013). *Harnessing constraint programming for poetry composition*. Proceedings of the international conference on computational creativity 2013, Sydney, Australia. (pp. 160–167).

Toivanen J. M., Toivonen H., Valitutti A., & Gross O. (2012). *Corpus-based generation of content and form in poetry*. Proceedings of the international conference on computational creativity 2012, Dublin, Ireland. (pp. 175–179).

Veale T. (2013). *Less rhyme, more reason: Knowledge-based poetry generation with feeling, insight and wit*. Proceedings of the international conference on computational creativity 2013, Sydney, Australia. (pp. 152–159).

Wiggins G. (2006). Searching for Computational Creativity. *New Generation Computing, Computational Paradigms and Computational Intelligence. Special Issue: Computational Creativity*, 24(3), 209–222.