Evolutionary Elaboration of Daily News as a Poetic Stanza

Pablo Gervás

Universidad Complutense de Madrid,Spain,
pgervas@sip.ucm.es,
WWW home page: http://nil.fdi.ucm.es

Abstract. Among the existing efforts at automatic poetry generation, either from scratch or from a given semantic input, little effort has been paid to the issue of sustained innovation. Sustained innovation requires that repeated attempts at generation produce results significantly different from prior system output. This is a characteristic of human creativity that is difficult to achieve in automatic systems built on deterministic procedures. The present paper describes a system that aims at sustained innovation by interleaving with the generation process a stage of knowledge collection or renewal. It extends an existing evolutionary solution for poetry generation to cover the generation of poems about what has happened in the world on a given day, taking as a knowledge source the set of news articles from a specific online newspaper for the given day. The collection of the newspaper articles for the day enacts the knowledge renewal stage. The system described creates a population of text drafts about the chosen day, and evolves them over a number of generations applying a fitness function that measures its metric and linguistic qualities, relying on operators designed to change the drafts with respect to the desired features.

1 Introduction

The issue of sustained innovation in creativity relates to the ability of an agent to produce significantly different results on a given generation attempt from those obtained earlier. In human creators this arises naturally from the fact that humans lead a life of exposure to continuous stimuli, and the knowledge base from which they feed their creative processes is continuosly updated with new material, whether from personal experience or from consumption of content rich products such as books, films, or conversations. For automatic systems, modelling of the creative processes in themselves is complex enough without attempting a parallel effort of modelling the processes of learning and interpretation that underlie such activities. Existing models of computer poetry generation tend to operate from a set knowledge base, usually a carefully crafted resource requiring significant development effort.

The present paper describes a poetry generator that is capable of producing a different set of outputs everyday, by using the set of newspaper articles for the day as a training knowledge base for its generation processes. The text for the newspaper articles is downloaded from the online edition of the newspaper. A language model is trained based on these texts and used to provide an initial population of drafts which are then submitted to an evolutionary process where they are repeatedly revised, shuffled and judged until an acceptable poetic elaboration of the initial material is reached.

Section 2 reviews briefly prior efforts at poetry generation. Section 3 outlines the evolutionary procedure employed and presents some results. Section 4 includes a discussion of the limitations of the resulting system as a model of sustained innovation in creativy, and section 5 presents conclusions and further work.

2 Brief Review of Existing Poetry Generation Work

Existing poetry generators are reviewed in terms of the specific AI technologies that are employed in the generation process.

2.1 Generate and Test

The generate & test paradigm of problem solving has also been widely applied in poetry generators. Because metric restrictions are reasonably easy to model computationally, very simple generation solutions coupled with an evaluation function for metric constraints are likely to produce acceptable results (given an assumption of poetic licence as regards to the content). An example of this approach is the early version of the WASP system [1]. Initial work by Manurung [2] also applied a generate & test approach based on chart generation, but added an important restriction: that poems to be generated must aim for some specific semantic content, however vaguely defined at the start of the composition process. This constitutes a significant restriction on the extent of poetic licence allowed.

2.2 Case-Based Reasoning

Another important tactic that human authors are known to use is that of reusing ideas, structures, or phrasings from previous work in new results. This is very similar to the AI technique of Case-Based Reasoning (CBR) [3]. Some poetry generators have indeed explored the use of this technique as a basic generation mechanism. An evolution of the WASP system [4] used CBR to build verses for an input sentence by relying on a case base of matched pairs of prose and verse versions of the same sentence. Each case was a set of verses associated with a prose paraphrase of their content. An input sentence was used to query the case base and the structure of the verses of the best-matching result was adapted into a verse rendition of the input. This constituted a different approach to hardening the degree of poetic licence required to deem the outputs acceptable (the resulting verses should have a certain relation to the input sentence).

2.3 Grammar-Based Generation

Another important mechanism that has been employed by automatic poets is grammar-based generation. By using a grammar to produce grammatically correct combinations of words, the results obtained start to resemble understandable sentences. As Chomsky mentioned in 1957 [5], the fact that a sentence is grammatically correct does not imply that it will be interpretable. However, in the context of automatically generated poetry, sentences like Chomsky's classic counterexample ("Colorless green ideas sleep furiously") acquire a special interest, as they provide both a sense of validity (due to their syntactic correctness) and a sense of adventure (due to the impossibility of pinpointing a specific meaning for them). On reading such sentences, the human mind comes up with a number of conflicting interpretations, none fully compatible with its literal meaning. This multiplicity of shifting meanings is very atractive in the light of modern theories about the role of reader interpretation in the reading process.

In 1984 William Chamberlain published a book of poems called "The Policeman's Beard is Half Constructed" [6]. In the preface, Chamberlain claimed that all the book (but the preface) had been written by a computer program. The program, called RACTER, managed verb conjugation and noun declension, and it could assign certain elements to variables in order to reuse them periodically (which gave an impression of thematic continuity). Although few details are provided regarding the implementation, it is generally assumed that RACTER employed grammar-based generation. The poems in Chamberlain's book showed a degree of sophistication that many claim would be impossible to obtain using only grammars, and it has been suggested that a knowledgeable combination of grammars, carefully-crafted templates and heavy filtering of a very large number of results may have been employed.

2.4 Stochastic Language Modelling

The use of n-grams to model the probability of certain words following on from others has proven to be another useful tecnique. An example of poetry generation based on this is the cybernetic poet developed by Ray Kurtzweil [7, 8]. RKCP (Ray Kurtzweil Cybernetic Poet)[9] is trained on a selection of poems by an author or authors and it creates from them a language model of the work of those authors. From this model, RKCP can produce original poems which will have a style similar to the author on which they were trained. The generation process is controlled by a series of additional parameters, for instance, the type of stanza employed. RKCP includes an algorithm to avoid generating poems too close to the originals used during its training, and certain algorithms to maintain thematic coherence over a given poem. Over specific examples, it could be seen that the internal coherence of given verses was good, but coherence within sentences that spanned more than one verse was not so impressive.

2.5 Evolutionary Solutions

Manurung went on to develop in his PhD thesis [10] an evolutionary solution for this problem (now described in [11]). Evolutionary solutions seem particularly apt to model this process as they bear certain similarities with the way human authors may explore several possible drafts in parallel, progressively editing them while they are equally valuable, focusing on one of them when it becomes better valued than others, but returning to others if later modifications prove them more interesting. Manurung's MCGONAGALL used a linguistic representation based on Lexicalized Tree Adjoining Grammar (LTAG) over which operated several genetic operators – from baseline operators based on LTAG syntactic operations to heuristic semantic goal-directed operators – and two evaluation functions – one that measured how close the solutions stress pattern was a a target metre, and one that measured how close the solutions propositional semantics was to the target semantics.

A redesigned version of WASP [12] was used to produce a selection of 10 poems which has been published in a book about the possibilities of computers writing love poems [13]. In that 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. The poems presented in the book were produced with content generators trained on collections of texts by Federico García Lorca [14], Miguel Hernández [15, 16] and a selection of Sixteenth Century Spanish poets [17]. Readers familiar with the sources can detect similarities in vocabulary, syntax and theme.

3 An Evolutionary Approach to Poetry Generation

The computational model proposed in this paper for the generation of poetry brings together two basic insights obtained from the study of the existing poetry generators: the ability to iterate over a draft applying successive modifications in search of a best fit, and the ability to measure metric forms. The concept of a *draft* that holds the current best solution for a given poem and which gets progressively modified towards an optimal solution, is fundamental to the proposed model. The concept of *reviser*, a module that operates on a draft to progressively improve it, completes the picture to cover the first insight. Such drafts need to be evaluated for conformance with the desired poetic form, and the results of this evaluation need to be taken into account in any subsequent operations on the draft. The concept of a *judge*, a module capable of evaluating partial results according to desired criteria, covers the second insight. In the model, judges can evaluate aspects concerning form, but also content, linguistic validity, fluency, or innovation (in the form of similarity with previous known poems). As a third insight, the model builds on the idea that poets do work at the same time on several possibilities for completing a line, keeping options open to see which may match better with the rest of the poem. When computers are considered to take on an equivalent task, this approach can be taken a step further, so a poetry generator can not just work on one poem but write several at the same time. The model will therefore operate not on a single draft but over a *population* of candidate drafts.

The existence of a population of candidate solutions, that evolves over time as a result of operations carried out upon it, and that is evaluated based on specific criteria, conforms with the structure of an evolutionary solution, which is one of the candidate technologies to apply. However, our aim is to provide the means for bringing together a number of these technologies. We do this in two different ways. First, we allow a set of alternatives for the creation of the drafts in the initial population. To this end we introduce the concept of *babbler*, a module in charge of producing an initial draft. By allowing a population of babblers to produce the initial population, we introduce the possibility of relying on more than one technology to produce them. Grammar, ngram, or case based solutions can be included among the set of babblers. Second, we introduce a set of alternatives for operating upon the initial drafts, by allowing a population of revisers, possibly employing different technologies. Finally, to allow for revision operations specific to poetic form, we introduce the concept of a *poet*, a module in charge of transforming a draft with a view to matching a specific poetic form. In the spirit of the model, we allow a population of poets, to contemplate more than one possible target form.

The resulting set of elements constitutes 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. Judges evaluate what babblers produce, revisers modify it taking into account what the judges have said. Bad sequences are eliminated during pruning, not so bad ones are modified to make them better.

3.1 WASP Redesigned: Evolutionary Approach to Poetry Generation

A redesigned version of the WASP poetry generator has been built following the model described above. 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 n-gram models obtained from the set of news articles from a specific online newspaper for the given day.

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.

Additional judges used assign scores for:

- overall poem length (with respect to a desired value used as input)
- length of the verses in the poem (with respect to a desired value used as input)
- average length of sentences in the poem (set empirically at 30 words, to achieve acceptable final poems)
- acceptability of the placement of stressed syllables with respect to each verse (as established by metric tradition for the desired verse length)
- appearance in the poem of words that are metrically impossible to parse (foreign words that do not follow the syllable patterns of the Spanish language)

Poets operate mainly by deciding on the introduction of line breaks over the text they receive as input.

Revisers rely on scores assigned by judges to introduce changes to drafts. Modifications can be of several types: deletion of spans of text, substitution of spans for newly generated ones, word substitution, sentence elimination, and simple cross-over of fragments of poems to obtain new ones. Revisers act as mutator operators, taking an existing draft and changing it in some way. The set of revisers includes:

- dropping a sentence at random
- inserting an additional sentence (produced from the same trained language model) at the end of the draft
- shifting the position of the line breaks in the draft to produce a new draft with different distribution of words over lines
- replacing some words of the poem selected at random with valid replacements according to the ngram model

Because an initial draft produced by an n-gram based content generator is then processed many times over by poets and revisers, final results oscillate between surprising faithfulness to the sources and very radical surreal compositions.

The implementation described here is a very simple one that combines two of the technologies previously used by story generators (evolutionary solutions and stochastic language models).

3.2 Empirical Setting of Evolutionary Parameters

The evolutionary WASP system has a number of parameters that govern its evolutionary process. The main ones are the size of the population to consider and the number of generations to be built. To establish acceptable values for these parameters, a number of experiments was run with the set of newspaper articles from the EL País newspaper for 21/05/2013. The system was tested with values of 100, 50, 20 and 10 for the population size and values of 100, 50, 20 and 10 for the number of generations. For each possible configuration, the system was run and the average score of the final population was recorded. Results are reported in Table 1. The time taken by each run is also included.

Pop Size	# gen	Av Score	Time in ms.
100	50	81	3159934
100	20	77	1369584
100	10	73	1025167
50	100	83	3308220
50	50	80	1624226
50	20	78	806429
50	10	74	490583
20	100	82	1505449
20	50	80	1028309
20	20	78	401746
20	10	75	279184
10	100	83	1337811
10	50	77	697729
10	20	79	264351
10	10	77	209875

Table 1. Average scores for different configurations of evolutionary parameters

The scores obtained increase steadily with the size of the populations and the number of generations. However, the time taken increases even more rapidly. As runs with 100 generations for a population of 50 or even 20 drafts result in comparable or higher scores than runs with a population of 100 drafts over 50 generations, and in less time (almost half in some cases), it was decided to carry out further experiments with smaller populations and larger number of generations.

3.3 Searching for Aesthetic Value

It is important to note that none of the judges included scores the poem for aesthetic value, or for grammaticality. The resulting population produced for any given run can be explored in search of acceptable poems. The construction procedure based on a language model trained on human-produced texts guarantees grammaticality to a certain extent, at least to ensure that the poems come up with sentences that sound acceptable in most cases. Those drafts that have suffered a high number of interventions (mutations) by the reviser that replaces words show less grammatical sentences. The interaction between revisers and judges guarantees that the poems in the final population satisfy metric constraints on verse length and stress placement according to traditional metrics.

Users of the system can navigate over the resulting population to select a poem that they find pleasing.

The system includes a number of parameters that can be employed to guide the generation process towards specific types of results, in terms of poem length, or verse length. Poem lengths considered are 14, 8 and 4, which are traditional values for classic stanzas of Spanish poetry (*sonetos*, *octavas reales*, and *cuartetos*). Verse lengths considered are 11 and 8 syllables, which are the traditional meters for Spanish poetry (*endecasílabos* y *octosílabos*).

Some examples of these poems, resulting from training with news articles from the EL País newspaper for 21/05/2013, are given in Table 2 and Table 3.

Valdano. Nosotros. Mourinho le había	*Valdano. Us. Mourinho had given him*
unos alumnos había	*some students had*
hecho música pero ambos chiítas	*made music but both Shiites*
los procedimientos sancionadores	*the sanctioning procedures*
y de cómo se apuntó	*and of how he claimed*
una mancha de justicia.	*a stain of justice.*

Table 2. Example of a poem produced by WASP after training on a collection of news articles from the EL País newspaper for 21/05/2013, with an approximate English translation.

Tengo nada que figuran	*I have nothing that figures*
con nuestra cultura es un	*with our culture is a*
laboratorio financiado	*laboratory financed*
con preferentes está	*with prefered (shares) is*
convirtiendo cada año.	*converting each year.*

Table 3. Example of a poem produced by WASP after training on a collection of news articles from the EL País newspaper for 21/05/2013, with an approximate English translation.

The poem in Table 2 resulted from a run of 10 generations with a population of 50 drafts, aiming for poems of 8 verses 8 syllables long. It obtained a score of 74 and it was rated 23rd of its generation.

The poem in Table 3 resulted from a run of 10 generations with a population of 50 drafts, aiming for poems of 8 verses 8 syllables long. It obtained a score of 75 and it was rated 18th of its generation.

4 Discussion

The poetry generator described here combines an evolutionary solution to model the creative process and a very simple model of the process of renewal of the source of inspiration. In humans, inspiration may come from an innumerable set of possible sources, from direct personal experience to material that has been read, or seen, or heard. The material employed may have been experienced since the last attempt at generation, or simply remembered from an earlier experience. To deal with this vast set of possible sources, humans apply filtering criteria, selecting from the whole set of available material just a small portion which is used to trigger the generation process. Additionally, humans can select different subsets of their remembered material and combine them in surprising ways to produce new results. The model of poetry generation presented here is significantly simpler in that it only considers a single set of resources: the set of texts that it employs as an inspiring set, in the sense described by Ritchie [18]. No selection is carried out over it, and renewal is radical: when the new texts for a day are loaded in the system all texts considered previously are abandoned. This is a limitation of the model.

Improvements can be considered along the lines of more elaborate processes of modification of the inspiring set, possibly by combining new material with the old, possibly by applying a process of selection to decide which parts of the old material are to be combined with the new.

An additional alternative might be to consider an inspiring set of texts that changes dynamically over time, influenced by system outputs, as described in [19].

5 Conclusions and Further Work

A computational model for poetry generation has been proposed that relies on evolutionary approaches to produce poems after training on a set of texts. This set of texts can be varied every day to emulate the way in which humans produce poetry inspired by different material on each successive ocasion that they set out to write. The system has been tested on the set of news articles from the digital edition of a Spanish daily newspaper for a given day, producing a set of poetic elaborations on the topics that are newsworthy on that day.

In many cases, pleasing poems ocur at the lower end of the scores in the population. This suggests that pleasing poems may be culled early during the evolutionary process. As future work, a judge capable of scoring poems on aesthetic potential may be explored. As aesthetic is a subjective quality, this task has significant inherent difficulty.

Acknowledgements

I would like to thank the referees for their comments which helped improve this paper. The research reported in this paper was partially supported by the Ministerio de Educación y Ciencia (TIN2009-14659-C03-01).

References

1. Gervás, P.: WASP: Evaluation of different strategies for the automatic generation of spanish verse. In: Proceedings of the AISB-00 Symposium on Creative & Cultural Aspects of AI. (2000) 93–100
2. Manurung, H.M.: Chart generation of rhythm-patterned text. In: Proc. of the First International Workshop on Literature in Cognition and Computers. (1999)
3. Aamodt, A., Plaza, E.: Case-based reasoning; foundational issues, methodological variations, and system approaches. AI COMMUNICATIONS **7**(1) (1994) 39–59
4. Gervás, P.: An expert system for the composition of formal Spanish poetry. Journal of Knowledge-Based Systems **14**(3–4) (2001) 181–188
5. Chomsky, N.: Syntactic Structures. Mouton, The Hague (1957)
6. Chamberlain, W.: The Policeman's Beard is Half Constructed. Warner Books, New york (1981)
7. Zacks, R.: The story of the 21st century - TR Associate Editor Rebecca Zacks interviews Ray Kurzweil. MIT Technology Review (2000)
8. Riordan, T.: The muse is in the software. The New York Times, 24 November (2003)
9. Kurzweil, R.: Ray Kurzweil's Cybernetic Poet. http://www.kurzweilcyberart.com/poetry/rkcp_overview.php Last visit: 7/03/2013 (2001)
10. Manurung, H.M.: An evolutionary algorithm approach to poetry generation. PhD thesis, University of Edimburgh, Edimburgh, UK (2003)
11. Manurung, R., Ritchie, G., Thompson, H.: Using genetic algorithms to create meaningful poetic text. J. Exp. Theor. Artif. Intell. **24**(1) (2012) 43–64
12. Gervás, P.: Computational modelling of poetry generation. In: Proceedings of the AISB 13 Symposium on AI and Poetry. (2013)
13. Gervás, P.: Diez poemas emocionales generados por un computador. In Cañas, D., González Tardón, C., eds.: ¿Puede un computador escribir un poema de amor?, Editorial Devenir (2010) 189–196
14. Lorca, F.: Poema del cante jondo ; Romancero gitano. Mil letras/ Thousand Letters. Ediciones Cátedra (2009)
15. Hernández, M.: Poemas de adolescencia: Perito en lunas ; Otros poemas. Biblioteca contemporánea. Editorial Losada (1963)
16. Hernández, M., Ballesta, J.: Viento Del Pueblo: Poesía en la Guerra. Letras Hispánicas. Cátedra (1989)
17. Rivers, E.: El Soneto español en el siglo de oro. Nuestros Clasicos Series. Akal (1993)
18. Ritchie, G.: Some empirical criteria for attributing creativity to a computer program. Minds & Machines **17** (2007) 67–99
19. Gervás, P.: Dynamic inspiring sets for sustained novelty in poetry generation. In: Proceedings of the International Conference on Computational Creativity 2011. (2011)