Computational Modelling of Poetry Generation

Pablo Gervás [1]

Abstract. Poems are particular well-crafted formulations of certain messages that satisfy constraints on their form as well on their content. From an engineering point of view, the ability to optimize the combination of form and content is a very desireable feature to be able to model. The present paper reviews a number of efforts at modelling poetry generation computationally, and it brings together insights from these efforts into a computational model that allows integration of a number of AI technologies combined together according to control structures compatible with observed human behaviour. The relation of this computational model with existing cognitive models of the writing task, and with models of computational creativity is discussed.

1 Introduction

Poetry is known to be a very advanced form of linguistic communication. Poems are particular well-crafted formulations of certain messages that satisfy constraints on their form as well on their content. From an engineering point of view, the ability to optimize the combination of form and content is a very desireable feature to be able to model. In the context of ongoing research as to how best present information available to computers, where the use of text is loosing ground under pressure from video and audio, the study of the phenomenon of poetic composition has a potential for pushing the frontiers of what computers can do in terms of presenting information as text in appealing ways.

As always, such an endeavour is faced with two possible approaches: to try to mirror the behaviour observed in humans, or to search for the best possible solution that available technologies might provide for this particular problem. The achievement of flight by man is often used as an example of how engineering practice may lead to the successful emulation of behaviours observed in nature. It is also used to illustrate the idea that the best emulation (such as a jet plane) of a natural phenomenon (such as the flight of birds) need not always mirror faithfully all the features of the inspiring phenomenon. In the case of poetry generation, the preferred approach has always been to exploit known techniques for modelling linguistic behaviour, rather than close study of human performance. Although some efforts have been made to develop models of the task of writing in general, to my knowledge no effort has been made to model computationally the specific task of writing poetry. The present paper reviews a number of efforts at modelling poetry generation computationally, and it attempts to bring together insights from these efforts into a computational model that allows integration of a number of AI technologies in combination. The relation of this computational model with existing cognitive models of the writing task, and with models of computational creativity is discussed.

[1] Universidad Complutense de Madrid, Spain, email: pgervas@sip.ucm.es

2 Brief Review of Poetry Generation Work

Existing poetry generators are reviewed from two different points of view: in terms of how they represent the fundamental elements that they combine, and in terms of the specific AI technologies that are employed in the generation process.

2.1 Form, Content and Articulation

Poetry generation systems explore a conceptual space characterised by form and content. The rules of the language being employed interconnect these two dimensions, in as much as any specific content, when phrased in a particular way in a particular language, thereby acquires a particular form. In an attempt to simplify the problem, AI systems wishing to emulate a poet's ability typically establish starting constraints on the output by restricting the exploration to a small subset of the search space.

These constraints can take the form of searching for poems in a particular stanza [11, 19, 20], starting from a restricted vocabulary [11], constraining sentences to satisfy a particular grammar, establishing restrictions on the semantics of the poem sought [19, 20], or combinations of these or similar constraints [19, 20].

Before the AI community got interested in poetry generation, there had been attempts to devise procedures for the systematic construction of poetry. Starting from a different background, generally closer to the humanities and to poetry itself, these initiatives applied a similar procedure to reduce the complexity of the problem, but relied on different ways of breaking down the problem into simpler elements. Some of these systematic procedures limit themselves to selecting a particular textual template with which the poems are produced [2], or reusing a predetermined set of verses [23].

2.1.1 Design of Generative Procedures

The designer of a generative procedure for any particular artifact needs to define some way of understanding the desired type of artifact in terms of properties it must satisfy, a structure it must follow, or ingredients that may be used in its construction [18, 6]. When the artifact is for a particular purpose, the designer may during this process focus on particular elements that are more relevant to this purpose, and pay less atention to other elements. As a result of these choices, the complexity and the versatility of the resulting generative procedure are affected. This is a well known problem in Natural Language Generation, where system designers have a broad range of options, from reusing canned text if there is just a small set of messages to be conveyed repeatedly, relying on templates for messages structure to be filled in with appropriate terms in each instance, or devising a more elaborate characterization of the subset of language to be generated if better coverage and fluency are desired [24]. This

initial analysis of the target artifact with a view to selecting a particular frame for understanding and decomposing it into parts that can later be used to assemble equivalent instantiations of the same type I will call *articulation*. This captures the concept of different parts being joined together in a whole, but also covers the concept of allowing the parts to move with respect to one another, and even the concept of appropriately conveying a desired meaning.

As every child knows who has ever owned an articulated toy, articulation has its limitations. Regardless of what the advertising said, a child's ingenuity will very soon come up with a particular pose that the available articulation cannot manage. And articulation comes at a price. Your Lego bricks will allow you to shift at will from car to plane to boat and back to car again, but every one of those shifts will require an effort in redesign (that some children would have preferred to spend playing).

Generative AI systems also suffer from these two problems: however refined your choice of representation for your problem, there will always be target artifacts that cannot be described are difficult to describe in it, and the more representational options you add to your system, the more complex its actual operation will have to be.

2.1.2 Articulation in Automated Poetry Generation

In the case of poetry generation the problem of articulation is compounded by the importance attributed to the form of the resulting text, added on top of the underlying complexity of language. This opens up two possible approaches to defining the understanding of poetry: from the point of view of language (grammar, vocabulary, semantics) and from the point of view of poetic form (stanzas, verses).

Depending on the degree of articulation of the generation procedure, some systems limit themselves to selecting a particular textual template with which the poems are produced, starting from a limited vocabulary, reusing a predetermined set of sentences or verses.

The degree of articulation captures the idea of how fine grained the representation used for content and form is in each case. Content can be considered simply at the level of texts (different texts have different content) or at an additional semantic level (a semantic representation is used for meaning of a given text, which allows different texts to have the same meaning). Form has historically been considered at many different levels: as metric restrictions on the output (stress patterns and length in syllables for verses, number and length of verses for stanzas), as poem templates to be filled [2, 7], as sets of verses to use [23], as sets of lexical items to use [11], as a language model to follow (obtained from a reference corpus) [15]. It is clear that many of these ways of restricting form carry an associated set of restrictions on content.

2.2 Techniques Employed for Poetry Generation

The review presented below attempts to cover some of the basic techniques that have been used as underlying technologies.

2.2.1 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 [10]. Initial work by Manurung [19] 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.

2.2.2 Evolutionary Solutions

Manurung went on to develop in his PhD thesis [20] an evolutionary solution for this problem (now described in [21]). 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.

2.2.3 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) [1]. Some poetry generators have indeed explored the use of this technique as a basic generation mechanism. An evolution of the WASP system [11] 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).

2.2.4 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" [4]. 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.

2.2.5 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 [31, 26]. RKCP (Ray Kurtzweil Cybernetic Poet)[15] 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.

3 Poetry Generation as a Mapping Effort

To my knowledge, none of the poetry generators described above was intended as a model of the human ability to generate poetry. Yet they provide a significant sample of human abilities related with linguistic creativity that have been modelled with reasonable success. These include: the ability to iterate over a draft applying successive modifications in search of a best fit, the ability to measure metric forms, the ability to reuse the structures of texts we liked in the past, the ability to rely on grammars for generating valid text, and the ability to use n-grams to produce a stream of text with surface form in a certain style. This list of abilities is doubtless not exhaustive, but it covers a broad range of aspects. The important idea is that although existing systems have identified and modelled these abilities, very few have considered more than one or two of them simultaneously. And yet intuition suggests that human authors are likely to apply a combination of all of these (and probably many more additional ones that have not been modelled yet) even in their simplest efforts.

It may pay to look in more detail at the set of tools that we have identified, with a view to considering how they might be put together in a single system if we felt so inclined. In doing this, we would be acting as Admiralty cartographers collecting sketches from various explorers, trying to piece together a map that accounts for all of them in a single representation.

3.1 A Frame for a Map: the Model Described

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.

3.2 A Draft Map: WASP Redesigned

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 mostly n-gram based. The poems presented in the book were produced with content generators trained on collections of texts by Federico García Lorca [17], Miguel Hernández [13, 14] and a selection of Sixteenth Century Spanish poets [27]. Readers familiar with the sources can detect similarities in vocabulary, syntax and theme.

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.

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.

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 only two of the technologies previously used by story generators (evolutionary solutions and stochastic language models), but it serves as an example of the kind of flexibility and articulation that the model allows.

This redesigned version of WASP was used to produce a selection of 10 poems which has been published in a book about the possibilities of computers writing love poems [12]. An example of one of these poems, resulting from training with poems by Miguel Hernández, is given in Table 1.

Odio vida, cuánto odio. Sólo	I hate life, how much hate. Only
por tu audición se ha desangrado.	by your hearing has it bled to death.
Ay de mi índice! Oh limón	Alas, my index! Oh, yellow
amarillo! Me darás	lemon! you will give me
un minuto de mar, vida	a minute of sea, life
como de alpistes, la tierra	as if made of bird seeds, the earth
que no dejarán desiertos.	that will not leave them deserted.
Ni las halles, guardalas	Do not even find them, put them away
en dos cajitas, hermano, como	in two little boxed, brother, as if
para niñas blancas.	for white girls.

Table 1. Example of a poem produced by WASP after training on a collection of poems by Miguel Hernández, with an approximate English translation.

This example, not originally included in the published collection, shows many of the peculiarities that arise from the method employed in production. Although the target metre was verses 8 syllables long (*octosílabos*), verses 1 and 2 are longer, because the words chosen by the babbler could not be cut otherwise by the poet. Verse 9 is stranger, because it is longer than desired although there is an alternative breaking that would have better satisfied the metre. The problem with the current set up is that it is very difficult to trace why particular results emerge as they do. For instance, this particular poem results from a sequence of interventions by different actors, summarised in the following trace signature that comes attached to the poem:

```
Babbler(Miguel Hernandez),
ParametrisedPoet(8,24),
LineBreakManager.recomputeLineBreaks8,
LineBreakJudgementShifter,
LineBreakManager.recomputeLineBreaks8,
SentenceDropper,
LineBreakManager.recomputeLineBreaks8,
LineBreakJudgementShifter,
LineBreakManager.recomputeLineBreaks8
```

This means that the text originally produced by the babbler (`Babbler(Miguel Hernandez)`) was distributed into 8 syllable verses (`ParametrisedPoet(8,24)`), then line breaks where shifted (`LineBreakJudgementShifter`), then one of the original sentences was dropped (`SentenceDropper`), then line breaks were shifted again (`LineBreakJudgementShifter`), with intervening stages where the suitability of the resulting verses with respect to the meter was tested (`LineBreakManager.recomputeLineBreaks8`). This example poem shows how the balance between form and content is taken to the edge, with almost correct metrical form (but with a few transgressions) and just enough grammaticality to allow some possible interpretation, while at the same time bringing words together in surprising combinations. This balance is due to the use of ngrams as an articulation choice, because they provide very tight local coherence between adjoining words and surprising freedom for other words in teh sentence beyond the window of specification of a single ngram.

4 Discussion

The proposed model presents a number of features that need to be discussed: the relation of the model with existing models of the writing task, and the relation with existing models of computational creativity.

4.1 The Map and Existing Models

It would be interesting to consider whether the model proposed in this paper can be related to existing models of the process of writing that have been collected by researchers in psychology and cognitive science, and researchers on the task of writing. Two such models are described below, to provide a basis for comparison.

4.1.1 Models of the Writing Task

Flower and Hayes [9] define a cognitive model of writing in terms of three basic process: planning, translating these ideas into text, and reviewing the result with a view to improving it. These three processes are said to operate interactively, guided by a monitor that activates one or the other as needed. The planning process involves generating ideas, but also setting goals that can later be taken into account by all the other processes. The translating process involves putting ideas into words, and implies dealing with the restrictions and resources presented by the language to be employed. The reviewing process

involves evaluating the text produced so far and revising it in accordance to the result of the evaluation. Flower and Hayes' model is oriented towards models of communicative composition (such writing essays or functional texts), and it has little to say about poetry. Nevertheless, a computational model of poetry writing would be better if it can be understood in terms compatible with this cognitive model. An important feature to be considered is that the complete model is framed by what Flower and Hayes consider "the rhetorical problem", constituted by the rhetorical situation, the audience and the writer's goals.

Sharples [28] presents a description of writing understood as a problem-solving process where the writer is both a creative thinker and a designer of text. He provides a description of how the typical writer alternates between the simple task of exploring the conceptual space defined by a given set of constraints and the more complex task of modifying such constraints to transform the conceptual space. Constraints on the writing task are described as "a combination of the given task, external resources, and the writer's knowledge and experience". Apparently the human mind is incapable of addressing simultaneously these two tasks of exploring within a set of constraints and modifying the set of constraints. Sharples proposes a cyclic process moving through two different phases: engagement and reflection. During the engagement phase the constraints are taken as given and the conceptual space defined by them is simply explored, progressively generating new material. During the reflection phase, the generated material is revised and constraints may be transformed as a result of this revision. Sharples also provides a model of how the reflection phase may be analysed in terms of specific operations on the various elements. During the reflection phase, the generated material is revised in a three step process of reviewing, contemplating and planning the result. During reviewing the result is read, minor edits may be carried out, but most important it is interpreted to represent "the procedures enacted during composition as explicit knowledge". Contemplation involves the process of operating on the results of this interpretation. Planning uses the results of contemplation to create plans or intentions to guide the next phase of engagement.

4.1.2 Comparison with Proposed Model

From a cognitive point of view, the set of operations postulated for the task of poetry generation aligns reasonably well with the processes described by Flower and Hayes. In terms of Flower and Hayes' model, operations of the planning process could be seen as taking place within the babblers (in planning the initial text) and in the poets (in planning the poetic form of the resulting draft). The operations of the translation process would in the simplest case be encapsulated within the babbler modules. However, it might be possible to envisage a more complex set of actors, including for instance a decomposition of a babbler into a pipeline of *planner* plus *translator*. A planner would then produce an intermediate representation (a poem plan?) which a translator would render into text. Such a solution would match existing approaches to analysing the natural language generation task [24].

The judges and revisers would be in charge of the processes of evaluation and revision would correspond to the reviewing process of Flower and Hayes' model. The role of the monitor, which allows and controls interaction between the various processes would here be represented by the overall evolutionary pattern of control.

In terms of Sharples' description of the writing task, the operations carried out by the babblers (and possibly poets) described in the model in this paper would take place during the engagement part of the cicle, and, the operations of judges and revisers would correspond to the reflection stage.

4.1.3 AI Techniques and Models of Writing Task

The importance given to planning in Flower and Hayes' model puts the spotlight on an AI technology that has so far not been applied to the task of poetry generation. This is in marked contrast to story generation, where the importance of causal relations in narrative comprehension has led to AI models of plot generation that rely heavily on the concept of planning. Many existing storytelling systems feature a planning component of some kind, whether as a main module or as an auxiliary one. TALESPIN [22], AUTHOR [8], UNIVERSE [16], MINSTREL [30] and Fabulist [25], all include some representation of goals and/or causality, though each of them uses it differently in the task of generating stories. As described above, the proposed model could be extended to include a planning component. Additionally, higher level planning processes, in the shape of a more intelligent monitoring process, could be set in place to automatically govern the configuration of the various populations of experts to involved in a particular generation process, or setting the various evolutionary parameters.

The engagement and reflection model [29] provides a useful framework to understand the proliferation of different technologies used for poetry generation. Sharples' concept of engagement seems to correspond with the ability to generate a new instance of a given artefact, without excessive concern to the quality or fitness for purpose of the partial result at any intermediate stage of the process. According to this view, case-based reasoning, grammars, or n-gram models can provide reasonable implementations of procedures for engagement. The concept of reflection captures the need to evaluate the material generated during engagement. Abilities like measuring metric forms would clearly have a role to play during reflection. However, it is important to consider that we are looking at a number of possible mechanisms for use in engagement, together with a number of possible mechanisms for use in reflection. This does indeed have a place in the general scheme proposed by Sharples. The evidence that we have presented so far suggests that a specific mechanism (or maybe more than one) may have been chosen to be used during a particular cycle of engagement. The process of reviewing mentioned by Sharples might simply be one of explicitly considering the choice of mechanism to use in engagement. The process of contemplating might be an application of the full set of evaluation mechanisms particular to reflection. The process of planning could be a complex process which would include among other things a decision of whether to change the engagement mechanism in use (or the configuration of any parameters it may need), and which mechanism to apply in each situation.

The model described allows a similar flexibility in the application of different mechanisms, without a need for specific control decisions to switch specific modules on or off. The evolutionary setting allows a set of different modules to compete in the generation process, with the best results as evaluated by the judges being chosen as solutions in the end over less valuable alternatives that may have been produced by less succesful tecniques.

4.2 Articulation and Creativity

Many efforts over the recent years that address the study of creativity from a computational point of view acknowledge as a predecessor

the work of Margaret Boden [3]. Boden proposed that artificial intelligence ideas might help us to understand creative thought. One of Boden's fundamental contributions was to formulate the process of creativity in terms of search over a conceptual space defined by a set of constructive rules. Sharples [28, 29] brings together Boden's computational analysis of creativity with insights on the task for writing, understood as a problem-solving process where the writer is both a creative thinker and a designer of text. The account instantiates the various elements in Boden's analysis as ingredients in the domain of the writing activity. For Sharples, the universe of concepts that can be explored in the domain of writing could be established in a generative way by exhaustively applying the rules of grammar that define the set of well-formed sentences. The conceptual space on which a writer operates is a subset of this universe identified by a set of constraints that define what is appropriate to the task at hand. These constraints limit "the scope of search through long term memory to those concepts and schemas that are appropriate to the task" [17 : p. 3]. Sharples identifies creativity in writing with the application of processes that manipulate some of these constraints, thereby exploring and transforming the conceptual space that they define.

The concept of articulation outlined at the beginning of the paper may help to refine these accounts of creativity in the particular context of poetry generation. The process of analysis of the target poems, and the definition of a particular representation, which we have called articulation, can be seen as actually determining the conceptual spaces over which the system is going to search for solutions to the poetry generation task. Sharples' example of defining the universe of concepts in a generative way based on a grammar would correspond to a particular choice for articulation. If the selected articulation is based on ngrams, a different conceptual space would result. If templates, or complete verses are chosen as means of articulation, the resulting conceptual spaces would be more restricted. Search would be easier, but coverage of possible resulting poems would also be extremely reduced.

5 Conclusions and Further Work

A computational model for poetry generation has been proposed that allows combination of a number of AI techniques that have been used in the past for this task. The model is compatible with intuitions drawn from the task of writing poetry as carried out by humans, and covers reasonable well the fundamental aspects of existing cognitive models for the writing task. The model has been used in the redesing of the WASP poetry generator, leading to an implementation that has produced poems that have been accepted for publication in a book. Although the actual implementation was very simple and did not exploit the full possibilities of the model, it showed the feasibility of combining different technologies in the proposed way.

A number of possible extensions have been identified, including the introduction of planning as an additional technology for poetry generation, the decomposition of the content generation task into planning and translation subtasks, and the development of more complex control mechanisms along the lines of the monitor in Flower and Hayes' model.

ACKNOWLEDGEMENTS

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

REFERENCES

[1] Agnar Aamodt and Enric Plaza, 'Case-based reasoning; foundational issues, methodological variations, and system approaches', *AI COMMUNICATIONS*, **7**(1), 39–59, (1994).
[2] Oulipo (Association), *Atlas de littérature potentielle*, number vol. 1 in Collection Idées, Gallimard, 1981.
[3] M. Boden, *Creative Mind: Myths and Mechanisms*, Weidenfeld & Nicholson, London, 1990.
[4] W. Chamberlain, *The Policeman's Beard is Half Constructed*, Warner Books, New york, 1981.
[5] Noam Chomsky, *Syntactic Structures*, Mouton, The Hague, 1957.
[6] E. Colabella, 'Generative art philosophy - ars artium', **24**(3), 241–266, (2011).
[7] Simon Colton, Jacob Goodwin, and Tony Veale, 'Full-FACE poetry generation', in *Proceedings of the International Conference on Computational Creativity 2012*, pp. 95–102, (2012).
[8] N. Dehn, 'Story generation after tale-spin', in *Proc. IJCAI 1981*, pp. 16–18, (1981).
[9] L. Flower and J.R. Hayes, 'A cognitive process theory of writing', *College Composition and Communication*, **32**(4), 365–387, (1981).
[10] P. Gervás, 'WASP: Evaluation of different strategies for the automatic generation of spanish verse', in *Proceedings of the AISB-00 Symposium on Creative & Cultural Aspects of AI*, pp. 93–100, (2000).
[11] P. Gervás, 'An expert system for the composition of formal Spanish poetry', *Journal of Knowledge-Based Systems*, **14**(3–4), 181–188, (2001).
[12] P. Gervás, 'Diez poemas emocionales generados por un computador', in *¿Puede un computador escribir un poema de amor?*, eds., D. Cañas and C. González Tardón, pp. 189–196. Editorial Devenir, (2010).
[13] M. Hernández, *Poemas de adolescencia: Perito en lunas ; Otros poemas*, Biblioteca contemporánea, Editorial Losada, 1963.
[14] M. Hernández and J.C. Ballesta, *Viento Del Pueblo: Poesía en la Guerra*, Letras Hispánicas, Cátedra, 1989.
[15] Ray Kurzweil. Ray Kurzweil's Cybernetic Poet. http://www.kurzweilcyberart.com/poetry/rkcp_overview.php Last visit: 7/03/2013, 2001.
[16] M. Lebowitz, 'Story-telling as planning and learning', in *Proc. IJCAI 1983*, volume 1, (1983).
[17] F.G. Lorca, *Poema del cante jondo ; Romancero gitano*, Mil letras/ Thousand Letters, Ediciones Cátedra, 2009.
[18] John Maeda, *Design by Numbers*, MIT Press, Cambridge, MA, USA, 2001.
[19] H. M. Manurung, 'Chart generation of rhythm-patterned text', in *Proc. of the First International Workshop on Literature in Cognition and Computers*, (1999).
[20] H. M. Manurung, *An evolutionary algorithm approach to poetry generation*, Ph.D. dissertation, University of Edimburgh, Edimburgh, UK, 2003.
[21] Ruli Manurung, Graeme Ritchie, and Henry Thompson, 'Using genetic algorithms to create meaningful poetic text.', *J. Exp. Theor. Artif. Intell.*, **24**(1), 43–64, (2012).
[22] James R. Meehan, 'Tale-spin, an interactive program that writes stories', in *Proc. IJCAI 1977*, pp. 91–98, (1977).
[23] R. Queneau, *100.000.000.000.000 de poèmes*, Gallimard Series, Schoenhof's Foreign Books, Incorporated, 1961.
[24] E. Reiter and R. Dale, *Building Natural Language Generation Systems*, Cambridge University Press, 2000.
[25] M. Riedl and Michael Young, 'Narrative planning: Balancing plot and character', *J. Artif. Intell. Res. (JAIR)*, **39**, 217–268, (2010).
[26] Teresa Riordan, 'The muse is in the software', *The New York Times, 24 November*, (2003).
[27] E.L. Rivers, *El Soneto español en el siglo de oro*, Nuestros Clasicos Series, Akal, 1993.
[28] M. Sharples, 'An account of writing as creative design', in *The Science of Writing: Theories, Methods, Individual Differences, and Applications*, eds., C. M. Levy and S. Ransdell. Lawrence Erlbaum Associates, (1996).
[29] Mike Sharples, *How We Write: Writing As Creative Design*, Routledge, June 1999.
[30] Scott R. Turner, *Minstrel: a computer model of creativity and storytelling*, Ph.D. dissertation, University of California at Los Angeles, Los Angeles, CA, USA, 1993.
[31] Rebecca Zacks, 'The story of the 21st century - TR Associate Editor Rebecca Zacks interviews Ray Kurzweil', *MIT Technology Review*, (2000).