UNIVERSITY OF EXETER

# Artificial Intelligence and Poetry

Ruth Aylett and Greg Michaelson (editors)

AISB Convention 2013 • University of Exeter • 3rd–5th April, 2013

# Foreword from the Convention Chairs

This volume forms the proceedings of one of eight co-located symposia held at the AISB Convention 2013 that took place 3rd-5th April 2013 at the University of Exeter, UK. The convention consisted of these symposia together in four parallel tracks with five plenary talks; all papers other than the plenaries were given as talks within the symposia. This symposium-based format, which has been the standard for AISB conventions for many years, encourages collaboration and discussion among a wide variety of disciplines. Although each symposium is self contained, the convention as a whole represents a diverse array of topics from philosophy, psychology, computer science and cognitive science under the common umbrella of artificial intelligence and the simulation of behaviour.

We would like to thank the symposium organisers and their programme committees for their hard work in publicising their symposium, attracting and reviewing submissions and compiling this volume. Without these interesting, high quality symposia the convention would not be possible.

Dr Ed Keedwell & Prof. Richard Everson
AISB 2013 Convention Chairs

Published by
The Society for the Study of Artificial Intelligence and the Simulation of Behaviour
http://www.aisb.org.uk

ISBN: 978-1-908187-30-7

AISB 2013
Artificial Intelligence and Poetry

Contents


# Artificial Intelligence and the Simulation of Behaviour Convention 2013 (AISB 2013)
# Symposium on Artificial Intelligence and Poetry
## 3rd March, 2013
## University of Exeter

Poetry exemplifies fundamental aspects of creativity which, in the view of proponents of Lady Lovelace’s objection that computers can do nothing original, focus strongly the differences between machines and humans. Poetry also exemplifies hard problems of practical AI, in natural language generation, expressive speech and non-verbal behaviour. This makes it a particularly valuable domain for AI,

Mechanical though non-automated techniques of poetry production long precede computers. Perhaps best known are the 1950s Beat poets’ use of cut-ups, following from the 1920s Dadaist approaches of random selection of words.

Almost all subsequent automated poetry generation, while considerably more sophisticated, still relies on guided random selection of words or phrases to populate syntactic structures under different constraints.

Thus, in one of the earliest documented instances, Christopher Strachey’s 1951 program generated love letters on the Manchester Mark 1 computer [1], for example:

*LOVE DUCK*
*MY ADORATION HOPES FOR YOUR LUST. YOU ARE MY WINNING ARDOUR: MY SWEET INFATUATION: MY ADORABLE FANCY: MY WINNING SYMPATHY.*
*YOURS LOVINGLY*
*M. U. C.*

There seems to have been contemporaneous reporting of other experiments in computer generated text. In December 1953, the UK magazine *Punch* ran a (human) poem by M. H. Longson entitled *Parnussus Mechanized* [2] with the caption:

> *(An electronic brain capable of producing poetry has been invented by a French scientist)*

The first verse is:

> *Will raging Science never rest content*

*And cease to vex our peaceful human scene?*
*The ocean harnessed and the atom rent,*
*She threatens now the Poetry Machine.*

Above the poem, appear white coated boffins intent on fiendish equipment; below, poets queue at the Labour Exchange. The cartoons were drawn by E.H. Shepard, better known for his *Winnie the Pooh* and *Wind in the Willows* illustrations.

The1968 exhibition *Cybernetic Serendipity: the computer and the arts*, held at the Institute of Contemporary Arts in London, contained a major section on computer poetry and text. The ten articles in the catalogue [3] suggest considerable international interest in both the processes and aesthetics of automated poetry generation. From the brief accounts of the programs, they all seem to depend on random selection and permutation using templates.

The articles also suggest fruitful collaboration between computer scientists and poets, perhaps reflecting C. P. Snow's optimism that the "two cultures" of science and art could be reconciled in a "third culture" [4]. In this catalogue, the celebrated Scottish poet Edwin Morgan discusses simulated computer poems. He also presents translations of poems in Italian generated by a program from Nanni Balestrini.

Our Symposium, while considerably more modest than the 1968 exhibition, nonetheless shows on-going interest in computers and poetry. 5 papers were submitted of which 3 were selected, by the Programme Committee:

- Ruth Aylett, Heriot-Watt University, co-organiser;
- Simon Colton, Imperial College, London
- Pablo Gervas Universidad Computense de Madrid;
- Kevin Knight , University of Southern California;
- Ian McDonagh, widely published poet;
- Christopher Newall, University of Hull;
- Greg Michaelson, Heriot-Watt University, co-organiser;
- Catherine Pelachaud, CNRS;
- Geraint Wiggins, Queen Mary University, London.

[1] D. Link, *LoveLetters_1.0. MUC=Resurrection. A Memorial*. http://alpha60.de/loveletters/2009_zkm/
[2] *Punch*, December 2nd, 1953, p667.
[3] J. Reichardt (ed), *Cybernetic Serendipity: the computer and the arts*, Studio International, special issue, 1968.
[4] C. P. Snow, *The Two Cultures: and A Second Look*, Mentor, 1963.

# The Muses of Poetry - In search of the poetic experience

**Diana Arellano** and **Simon Spielmann** and **Volker Helzle** [1]

**Abstract.** This paper presents the current advances in "The Muses of Poetry", an on-going project that combines interaction, emotions and poetry. The goal of the project is to create an interactive installation where a virtual character not only recites poetry, but also manifests the affective content of the poem through facial expressions and voice rhythm. The novelty of our work is the combination of different elements like human computer interaction, semantic analysis, affective computing and character animation, which intend to create a unique poetic experience. These elements are presented in the first part of the paper. In the second part, we do a more in-depth analysis of all the aspects that would be necessary to have virtual characters that read poems out loud, while conveying their intrinsic emotional meaning.

## 1 INTRODUCTION

There is no doubt that poetry is one of the most creative forms of literary expression. A good poem is not only capable of eliciting mental images and awake feelings in the reader, but its rhythm and melody can transport this reader to the world created by the poet.

This is reflected in the words of Emily Dickinson, who defined poetry in the following way "If I read a book and it makes my whole body so cold no fire can warm me I know that is poetry. If I feel physically as if the top of my head were taken off, I know that is poetry. These are the only way I know it. Is there any other way?" (L342a, 1870).

However, one thing is to read poetry for oneself and another thing is to read it out loud. The latter presents a major challenge because the reader needs to be aware of the style of the poem, the pauses, the melody and the ideas that this poem intends to convey.

Having these elements in mind, we posed the question: is it possible to have a virtual character that reads poetry? Furthermore, can this virtual character make an audience connect with a poem, with its words and meanings?

Based on our previous experience with interactive virtual characters, we decided to tackle these questions by creating an interactive and audiovisual poetry installation. The work we present in this paper has been carried out within the framework of the on-going project "The Muses of Poetry", a media installation which main objective is to bring people closer to poetry.

One of the novelties of our work is the way poetry is addressed. It is not our intention to automatically generate poems, or just to have a character reading poetry. Instead, the installation allows users to interact with a virtual character who expresses the underlying affective content of each poem. Moreover, the project would also give poets from around the world the opportunity to expose their poems more publicly and to a wider audience.

[1] Institute of Animation, Filmakademie Baden-Wuerttemberg, Germany, email: diana.arellano@filmakademie.de

Another contribution is the mixture of different areas like semantic analysis, real-time computer graphics, voice generation and human-computer interaction, to create very differentiated virtual characters that engage the user in a poetic experience.

Nevertheless, the quest for more intelligent, affective and creative characters raises a number of issues from a technological and somehow philosophical point of view. In the following sections, we present the current state of our project as well as the aspects that need to be taken into account to provide our characters with intelligence and creativity.

## 2 RELATED WORK

The majority of the existent works that combine poetry and artificial intelligence deal with "how to generate written poetry" instead of "how to automatically read poetry". Examples of these are Colton et al. [4], who came up with a corpus-based poetry generation system that constructs poems according to a given rhyme, sentiment, word frequency and similarity; David Cope, who created the program "Alena" (Artificial Life Evolving Natural Affinities) to automatically write haiku, which were subsequently published in the ebook "Comes the Fiery Night" [5]; Pablo Gervás, who created WASP, a reasoning rule-based system that takes as input a set of words in Spanish and verse patterns and returns a set of verses [7]; or Toivanen et al. [19], who made used of text mining methods, morphological analysis, and morphological synthesis to generate poetry in Finnish.

Other works, more related to our research, are the ones of Tizhoosh and Dara [17] and Tizhoosh et al. [18], which focused on the analysis of text in order to distinguish between poem and prose, without understanding or interpreting the underlying poetic meaning. Similarly, there have been a number of researchers working on assessing the style of a poem, either to use it as a tool for the study of different types of poems and see how they affect the reader's perception of the poem [10], or to figure out what makes a poem beautiful [9].

As for the use of interactive virtual characters, Naoko Tosa was one of the pioneers in this area with her interactive installation "MUSE" [20]. In MUSE, poems are created by exchanging poetic phrases between the user and the system, represented by a character which facial features are eyes, eyebrows and mouth. In Tosa's work, the facial expressions of the character change according to the emotions conveyed in the phrases uttered by the user. In the same direction of interactive systems, Kwiatek and Woolner [12] merged poetry into interactive storytelling based on still and video panoramas. The aim of their application was to develop interest not only in the life of the poet Charles Causley but also in his literary output.

In the field of emotional speech synthesis, works like [2], [15], or [1] have tried to achieve expressiveness in the synthetic voice. Nevertheless, it still remains an open issue.

## 3 OVERVIEW OF THE SYSTEM

As we mentioned in Section 1, The Muses of Poetry is an interactive installation where a virtual character manifests the intrinsic emotionality of a poem through facial expressions and affective speech.

In this section, we present a brief technical overview of the implementation of the modules that are part of the system behind The Muses of Poetry (see Figure 1).

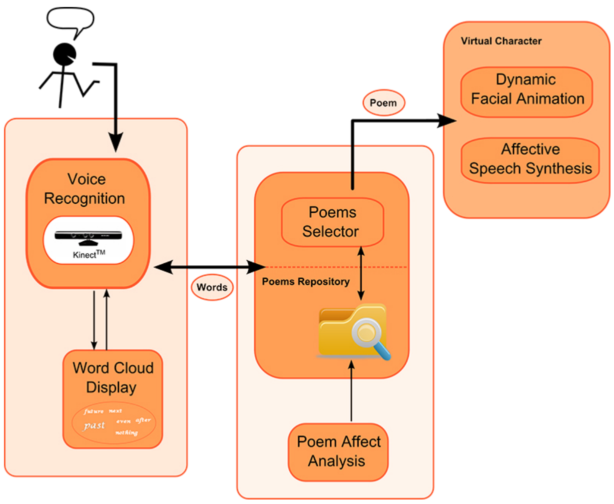


**Figure 1.** Modules of the system behind The Muses of Poetry

**Figure 2.** Example of interaction

### 3.1 Words Visualization and Voice Recognition

The interaction with The Muses of Poetry begins when the user says three words aloud, which are recognized by the system allowing it to select a poem that contains those three words.

The Words Visualization and Voice Recognition module uses and implements the elements that allow the interaction of the user with the system: a Microsoft Kinect, the Word Cloud Display submodule, and the Voice Recognition submodule.

The Kinect array microphone is the device that captures the voice of the user, which is sent as input to the Voice recognition submodule. The reason for using Kinect and not other type of microphone is to provide the user with a very natural way of interaction (i.e. free of cables as if in a person-to-person conversation).

In the current implementation of The Muses of Poetry, we use the Microsoft Speech Platform as the speech recognition engine. Therefore, we are able to define in the Kinect the *grammar*, or set of words that should be recognized by the system. The grammar consists of all the words from all the poems in the repository with a length over 3 characters. Thus words like “a”, “the”, “or”, and so on, would not be recognized.

In order to let the user know which words he or she can say, these are displayed in a “word cloud” arrangement, which is generated dynamically in every interaction. The first time the user approaches the installation a word cloud of the 30 most frequent words is shown. After a word is said and recognized, the cloud shrinks showing the most frequent words in those poems that contain the previously recognized word. This process is repeated once more, after which the control is passed to the Poem Selector module. Figure 2 shows an example of interaction with the installation.

### 3.2 Poem Affect Analysis

An important step that is performed prior to the interaction with the installation is the affective analysis of the poems. In principle, this is done just once for each poem that is added to the repository.

To extract the affective information of each poem we carry on a semantic analysis, aided with the Whissell’s Dictionary of Affect in Language (WDAL) [6]. This dictionary includes 10,368 English words with affective connotations, where each one is described with regard to the dimensions of Activation (or Arousal) and Evaluation (or Pleasantness). Two reasons led us to decide for the Whissell’s dictionary and not for others like ANEW [14] or WordNet-Affect [16]. First, Whissell’s dictionary has been created with words from literary and poetic texts. The other two contemplate mostly affective words, which are not enough to assess the whole meaning of a poem. Second, the number of words contained in Whissell’s dictionary is greater than in ANEW and Wordnet-Affect. As a result, the majority of words used in poems are mostly recognized by Whissell’s dictionary and not by the other two.

The WDAL itself operates as a licensed stand-alone application that assesses the affective information of a poem in terms of activation, evaluation and imaginary dimensions. It also provides a detailed classification of each word in the poem according to the following states: pleasant, nice, fun, passive, sad, unpleasant, nasty, active, high imagery and low imagery. Currently, we only consider the pleasant, nice, fun, sad, unpleasant and nasty states.

Nevertheless, if we associate emotions only to words, then it might happen that the emotional connotation of the poem is distorted. A way to avoid this is to tag the poem using a more general structure than the word, but smaller than a stanza (i.e. a unit within a larger poem). This would be the lines of the poem, which are the ones that convey the ideas the poet wanted to express. As a result, from the global analysis of all the words in the poem, we depict if it is emotionally positive or negative. In a second analysis, we break down the poem in lines. Then, the prevailing emotion in the words of a line is the one that is applied to it. For instance, if a line contained more unpleasant words and the poem was assessed as negative, then that line would be tagged as “unpleasant”.

After this process is completed, the poems with their corresponding emotional tags are stored in the repository.

### 3.3 Poems Selector

This module is the one that accesses the poem repository, and based on the words recognized from the user selects the poem to be performed.

The selection of the poems is done in a top-down approach. At first, the selector considers all the available poems in the repository, extracts their words and sends them to the Voice Recognition and Word Cloud module. When the first word is recognized, the selector searches for it in the current set of poems, keeping only the poems that contain that word. Moreover, the selected word is stored, so it does not appear again in the word cloud. This process is repeated twice more, guaranteeing that the poems in the final set contain the three recognized words.

Finally, if the final set contains more than one poem, one of them is picked randomly and sent to the Virtual Character module.

## 3.4 Virtual Character

This module has been divided into two submodules very closely related: the facial animation module and the speech synthesis module.

### *3.4.1 Dynamic Facial Animation*

One of the main characteristics of The Muses of Poetry is the dynamic generation of facial animations. In this way, there is no need of having pre-rendered animations for each poem. It represents a great advantage because any new poem can be added into The Muses without effort on the animation side. The development framework that provides this functionality is named Frapper[2], created at the Institute of Animation, Filmakademie Baden-Wuerttemberg.

In order to produce dynamic animations, first a set of pose animations are created in the third-party software Maya, and then exported to Frapper in the form of files containing the references to the object and its animation properties. Currently, these animations correspond to universal emotions like "anger", "sadness", "disgust" and "joy"; to idle states like "waiting" or "thinking"; or to "confirmation" of a recognized word. Being this the first prototype, we preferred to keep a reduced set of emotions, which can be enhanced in future iterations of The Muses. Figure 3 shows two of the expressions manifested by the current female character.

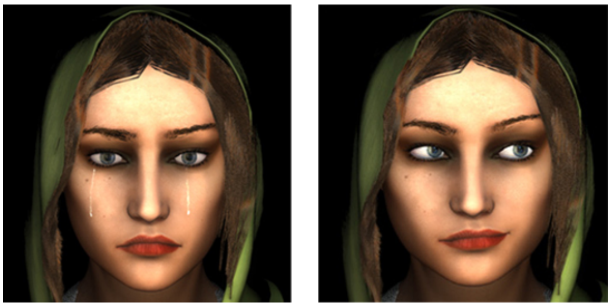

**Figure 3.** (Left) Expression of "sadness". (Right) Expression while "waiting"

To generate a poem animation, Frapper takes into consideration the emotional tags in the text of the poem. For each tag there is a number of associated animation files, which are randomly chosen by Frapper, providing variability to the poem animation.

Another important feature of Frapper is automatic lip-syncing. It is accomplished by using the visemes information provided by the speech synthesizer. Thus, having previously defined the poses for each of the selected visemes (i.e., A, O, E, CH, I, F, M), these are triggered when they are recognized in the text to be recited.

Additional rendering elements provided by the framework are tears simulation, eye redness, and wrinkles generation.

### *3.4.2 Speech Synthesis*

To create the voice of the character we used a third-party voice synthesizer, provided by SVOX[3].

It produces not only a more natural voice, but also the visemes information required by Frapper to generate automatic lip-sync. Similarly, it allows us to control the pitch and speed of the voice, which would produce variations that enhance the emotionality of the voice. In the current version of The Muses of Poetry these features are not yet exploited, consequently, the emotional content of the poem is manifested mostly through facial expressions.

# 4 Philosophical aspects of a Virtual Poetry Reader

As seen in the previous sections, The Muses of Poetry is a combination of multiple disciplines and technologies used to achieve an engaging poetic experience. Nevertheless, when showed to students and colleagues, a number of issues were pointed out regarding the intelligence and creativity of the character and the installation. Therefore, the question we asked ourselves is: what do we need to make our character more intelligent and creative?

After an in-depth research, we found that the answer to this question poses a number of issues. Even though they are planned for future work, they make us think of all the implications when creating an Affective Virtual Poetry Reader. These thoughts are discussed in the following subsections.

## 4.1 Perception

"*What people think when interacting with The Muses of Poetry?*"

This is the query that came to our mind when we showed it to an audience for informal testing purposes. It is worth noting that so far we have not performed any perceptual test that would give us qualitative and quantitative results, given that this is still a work in progress. However, we tried to show occasionally the installation to an audience formed mainly by animation and interactive media students, computer scientists and poets, in order to get their feedback.

The answers have been as varied as the audience that responded, showing the importance of perception when dealing with an installation of this kind. One of the more recurrent topics was the voice, which will be explained in a further subsection.

The visualization of the character was another important issue, with opinions regarding its abstraction, its expressiveness, or the way it should be displayed. The majority of the persons, including poets, reacted positively to the facial expressions, agreeing that it conveyed emotionality. Nonetheless, they were also distracted by the effect of what they called "robotic" voice. There was one case where a person manifested the opposite reaction. He found the voice completely appropriate but the expressions subtle.

From these comments, we realized that the expectations of regular users when faced with a realistic human character make them want to see and hear a real person reciting and expressing the affect in a poem. That leads to the problem that any artefact, however small, diminishes the whole experience.

[2] http://frapper.animationsinstitut.de/

[3] http://www.nuance.de/products/SVOX/index.htm

Given than the modification of the synthesized voice to make it sound human-like (i.e. with the nuances and expressiveness of natural voice) was not possible at this stage; and as a matter of fact, it remains a challenge in the Speech Synthesis field, we decided to change the facial appearance of the character. By moving away from a realistic character into a more abstract one, we wanted to see if the "robotic" voice could be accepted as part of the character, and not as a failure in the character.

To address this topic, we asked a group of students from the Filmakademie Baden-Wuerttemberg to come up with more abstract concepts of virtual characters. One of the designs is shown in Figure 4, a virtual head made of particles that swirl around, producing the effect of a constantly moving mask.

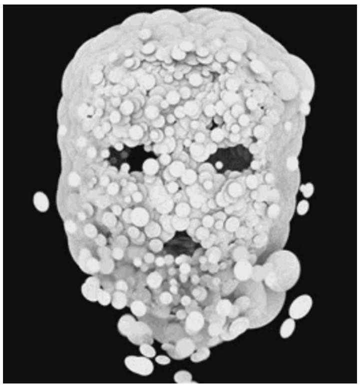

**Figure 4.** Concept design for an abstract character

The speech in this abstract character was made by simply opening and closing its mouth, according to the words of the recited poem. To have an idea of the effect that it would produce, we presented it to a small audience with experience in animation. Again the reactions where very dissimilar. On the one hand, some of them were fascinated by the swirling effect. On the other hand, some expected to see facial expressions during the interaction.

After this second experiment, we concluded that when working with abstraction and imagination, it is very difficult to use one type of character to target all audiences. Imagination and perception play a crucial role, hence the need to address these two points. One possibility in this direction would be to have a character that could change according to the user's perception of poetry, making it a first step into creating intelligent "adaptable" characters. In the near future, we will allow the participants to choose from a number of characters, in a way to personalize the experience.

## 4.2 Semantic Analysis

As explained in Section 3.2, currently we are using an affective dictionary to perform the semantic analysis of the poems and extract their affective content. Previous works have dealt with text analysis of poetry using different techniques, like statistical analysis to assess the poetry style of different poets [11], logic rules and pattern matching to extract metaphors from the poems [8], or classifiers like Bayesian and Multilayer Perceptron to extract poetic features to differentiate prose from poetry [18].

The main challenge in The Muses of Poetry is that all the poems are free verse. It means that they do not use a consistent meter pattern or rhyme, tending to look and sound like prose. Therefore, most of the techniques used in previous works result inefficient.

The next step would be to find a way to automatically extract syntactically well-formed sentences, and group them in a way so they convey one complete meaning. Then, we would need to teach the system how to recognize elements like the affective cues in it, and tag the text accordingly. Breaking the poem into its lines would also give the "space" needed to make the character pause. This would result in a more intelligent affective poetry analyser.

## 4.3 Poetic Voice

According to Rachel Blau DuPlessis (cited in [13]), poetry "is the kind of writing that is articulated in sequenced, gapped lines and whose meanings are created by occurring in bounded units... operating in relation to pause or silence". From this definition it can be inferred that poetry is sound. Hence the importance of identifying the lines in a free verse poem to know its periodicity and its gaps.

The question is then *how to make our character transmit the sound of poetry?* The answer again is not trivial, specially when there is not a specific way to read poetry to take as guidance. Each poet has its own style, which makes it more difficult for the system to learn a particular "reciting" pattern. If we take as example the style of poet William S. Burroughs or Robert Frost, we could hear that they had a characteristic way of accentuating the words, almost as if they "sing" the poem. However, there are other poets who read in a way that resembles the current style in The Muses of Poetry. One of the reasons might also be that the metrical feet of the poems in the installation is *Iamb*, which it is said is the nature of the English Language.

Another related issue is the emotionality in a computer generated voice. When it comes to emotions in speech, so far the majority of speech synthesizers in the market and in academia offer voices that are not yet expressive enough.

Buckhardt and Stegmann [2] identified the features of the speech signal as "spectral (sound of the voice), prosodic (melody of the speech), phonetic (kind of spoken phones, reductions and elaborations), ideolectal (choice of words) and semantic features". However, achieving speech with the right features that simulate human speech would require a great investment of time and resources. A proof of this is that despite the number of methods proposed for achieving expressive speech synthesis (e.g., rule-based, data-driven, among others), it still remains as a challenging research issue. Thus it is not our intention to create our own speech generator, but to be able to fine-tune parameters in existent ones that allow us to have a more natural voice with emotions.

A proposed solution was the use of human-recorded voices, but then it would go against the automation achieved in our system. The Muses of Poetry has been conceived as an installation where the character poses the expressions and generates the speech that is more adequate to certain poem.

Basic rules that could be applied to make our characters read poetry with more emotionality and vigour would be:

- Poems come in lines, but pausing at the end of every line will create a choppy effect and interrupt the flow of the poem's sense. Readers should pause only where there is punctuation, just as you would when reading prose, only more slowly [3].
- Contrary to the rule above, sometimes the pause in-between lines are actually pauses, but so short that is like the reader's breath inhale. A possibility would be to identify the correct spacing in-between lines pauses and simulate the inhaling sound.

- Another kind of pause would be before connectors: “and”, “but”, “or”; or after “so”. Nonetheless, this is not a fixed rule and does not happen always.
- Some poets recite the end of the line much slower to give emphasis to the meaning of the poem. This can be done with sad or happy poems, but not with unpleasant or funny poems, where probably a raise in the volume would produce the desired effect.

A concluding remark regarding speech synthesis has to do again with the issue of perception. When people see and interact with a realistic character, the expectations are high and any flaw in the representation is taken as an “uncanny” effect. But, what would happen if the character is so abstract that there is no way to link it to a known voice? Would the poetic effect be the same? Would the audience perceive it as a real intelligent entity that analyses the poem and manifests the emotions it conveys?

These are questions that remain open and we intend to investigate and test in the further development of The Muses of Poetry.

## 5 Discussion and Future Work

The Muses of Poetry is the name of the project that aims to create an interactive installation where emotional poetry “reciters”, or muses, transmit not only the words of a poem but also its intrinsic emotional meaning.

The innovative combination of human-computer interaction (HCI), media design, computer graphics, voice synthesis and semantic analysis resulted in the first prototype of The Muses of Poetry. This has helped us to assess which should be the path to follow in order to convey a very engaging poetic experience.

One of the first things we noticed was the way people perceived poetry. For some of them, poetry belongs to the realm of abstraction and imagination. Thus, their expectations when faced with a realistic character reading poetry made them find a number of flaws and elements that were not suitable. The voice of the character being too robotic, the animations being too subtle, or the lack of an environment that separates the audience from the real world, were part of the feedback of the users who tried the installation.

Another element to take into consideration is the need for a more “intelligent” character that would make each interaction unique. This could be achieved, for instance, by adding a mood element to the semantic analysis. Thus poems would be analysed differently depending on the mood of the character. Moreover, this mood could change depending on the interaction with the user, or the amount of sad, happy, or unpleasant poems the character has read so far.

Intelligence would also mean having a more emotional voice that goes according to the affective state of the poem. Currently, our prototype does not consider emotionality in the voice due to technical reasons, but it is intended for future work. The proposal is to link visual and vocal expression in such a manner that, independently of the character, gives the impression of a living entity reciting poetry.

As it can be seen, there is plenty of room for improvement. Many questions are left opened but are intended to be answered in future iterations. So far real-time rendering, automatic lip-sync, natural interaction, and affective semantic analysis have been invaluable tools that allowed us to come to this point. Now it is time to endure the task of giving intelligence to the system to achieve the ultimate goal of transporting the user to the world created by the poet and manifested in The Muses of Poetry.

## ACKNOWLEDGEMENTS

This work is funded by the *Innovationsfonds Kunst*, from the Ministerium für Wissenschaft, Forschung und Kunst (MWK) Baden-Württemberg. We also thank Tatiana Pushkareva for the animations of the female virtual character, Amit Rojtblat for the concept and animations of the abstract character, Andreas Hykade for his advises on the visual aspect of the abstract character, and Kent MacCarter for being our poet adviser.

## REFERENCES

[1] A.W. Black, H.T. Bunnell, Ying Dou, P. Kumar Muthukumar, F. Metze, D. Perry, T. Polzehl, K. Prahallad, S. Steidl, and C. Vaughn, ‘Articulatory features for expressive speech synthesis’, in *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, pp. 4005–4008, (2012).

[2] F. Burkhardt and J. Stegmann, ‘Emotional speech synthesis: Applications, history and possible future’, in *German Conference on Speech Signal Processing, ESSV*, (2009).

[3] B. Collins. How to read a poem out loud. ONLINE, 2004. http://www.loc.gov/poetry/180/p180-howtoread.html. Last checked: 01/25/2013.

[4] S. Colton, J. Goodwin, and T. Veale, ‘Full face poetry generation’, in *Proceedings of the Third International Conference on Computational Creativity*, (2012).

[5] D. Cope, *Comes the Fiery Night*, NY: Amazon Books, 2011.

[6] P. Duhamel and C. Whissell. The dictionary of affect in language [computer software], 1998.

[7] P. Gervás, ‘Wasp: Evaluation of different strategies for the automatic generation of spanish verse’, in *Symposium on Creative & Cultural Aspects and Applications of AI & Cognitive Science*, (2000).

[8] L. S. Hiyakumoto, *PoEM : a parser of emotion metaphors*, Master’s thesis, Massachusetts Institute of Technology. Dept. of Architecture. Program in Media Arts and Sciences, 1999.

[9] J. Kao and D. Jurafsky, ‘A computational analysis of style, affect, and imagery in contemporary poetry’, in *NAACL Workshop on Computational Linguistics for Literature*, (2012).

[10] D. M. Kaplan and D. M. Blei, ‘A computational approach to style in american poetry’, in *Proceedings of the 2007 Seventh IEEE International Conference on Data Mining*, ICDM ’07, pp. 553–558, (2007).

[11] David M. Kaplan and David M. Blei, ‘A computational approach to style in american poetry’, in *Proceedings of the 2007 Seventh IEEE International Conference on Data Mining*, ICDM ’07, pp. 553–558, (2007).

[12] K. Kwiatek and M. Woolner, ‘Let me understand the poetry. embedding interactive storytelling within panoramic virtual environments’, in *EVA 2010*, pp. 199–205, (2010).

[13] B. McHale, ‘Beginning to think about narrative in poetry’, *Narrative*, **17**(1), 11–27, (2009).

[14] F. A. Nielsen, ‘A new anew: evaluation of a word list for sentiment analysis in microblogs’, in *Proceedings of the ESWC2011 Workshop on ’Making Sense of Microposts’: Big things come in small packages*, eds., Matthew Rowe, Milan Stankovic, Aba-Sah Dadzie, and Mariann Hardey, volume 718 of *CEUR Workshop Proceedings*, pp. 93–98, (2011).

[15] M. Schröder, ‘Expressive speech synthesis: Past, present, and possible futures’, *Affective Information Processing*, 111–126, (2009).

[16] C. Strapparava and A. Valitutti, ‘Wordnet-affect: an affective extension of wordnet’, in *4th International Conference on Language Resources and Evaluation (LREC 2004)*, pp. 1083–1086, (2004).

[17] H. R. Tizhoosh and R. A. Dara, ‘On poem recognition’, *Pattern Analysis and Applications*, **9**(4), 325–338, (2006).

[18] H. R. Tizhoosh, F. Sahba, and R. Dara, ‘Poetic features for poem recognition: A comparative study’, *Journal of Pattern Recognition Research*, **3**(1), (2008).

[19] J. M. Toivanen, H. Toivonen, A. Valitutti, and O. Gross, ‘Corpus-based generation of content and form in poetry’, in *Proceedings of the Third International Conference on Computational Creativity*, (2012).

[20] N. Tosa, ‘Interactive poem’, in *ACM SIGGRAPH 98 Conference abstracts and applications*, SIGGRAPH ’98, pp. 300–, (1998).

# 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).

# Emotional and Multi-agent Systems in Computer-aided Writing and Poetry

**Alexis Kirke[1], Eduardo Miranda[1]**

**Abstract.** MASTER (Multi-Agent System for Text Emotion Representation) is an artificial society in which each member has a digital emotional state. Member agents attempt to influence each other's emotions by reciting "poems" to each other which express their own emotional state. As agents do this, larger texts are developed in the society through social learning. The resulting texts are not meaningful in the normal sense of everyday language - the sound and word repetition generates meaning. Like normal English, there is actually a hierarchical structure to the repetition (i.e. repetitions within repetitions), and the words are often evocative and sometimes contrasting.

## 1 INTRODUCTION

Some computer poetry research focuses on demonstrating the ability of a technique at simulating poetry, whilst others focus on assisting in the creative acts. This can be viewed as similar to the distinction in computer music between algorithmic composition and computer-aided composition [1]. Computer-aided composition is used as a form of digital collaborator between human and computer which can move the human composer into new areas of creativity, perhaps breaking them out of old habits. In this paper a computer-aided poetry system is introduced, MASTER (Multi-Agent System for Text Emotion Representation) [2]. MASTER is designed to investigate if a Multi-agent System which has no explicit knowledge of how language is constructed, can still help to generate emergent poetry. There has been work on MAS analysing poetry [3] and on MAS being used for story generation and character evolution in prose [4, 5]. As far as we are aware MASTER is the first generative poetry system utilizing multi-agent systems and artificial emotion.

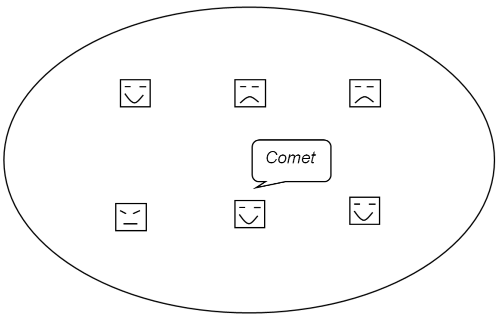


**Figure 1:** A Heuristic Representation of MASTER

Multi-agent systems (MAS) [6] are composed of multiple interacting intelligent agents. An agent is an autonomous entity which observes and acts upon an environment (i.e. it is an agent) and directs its activity towards achieving goals. Examples of problems which are appropriate to multi-agent systems research include online trading, disaster response, and modelling social structures. A key property of MAS is their ability to generate unexpected or novel responses to problems, sometimes called "emergence" [7]. They have been used successfully in computer-aided composition, because of their emergent properties.

## 2 COMPUTER POETRY

Common techniques in algorithmic and computer-aided poetry include words being chosen from a hand-crafted dictionary and inserted into a framework [8] (e.g. haiku or sonnet form). It is also possible to make a statistical language model based on existing poems or other texts – this incorporates information about which words / phrases follow which, and their frequency of occurrence [9]. A further approach is to create a set of rules for generating (or re-generating text) based on a manual or automatic analysis of other poetic text [10].

An example output of [9] (Kurzweil's "Cybernetic Poet") is shown below. It is called "Wondered" and is written after the system was trained on the poems of Dave Gitomer:

*today i wondered*
*if i mused*
*today i saw you*
*i learned*
*in awe and you*
*if i wondered*
*if i mused*
*today i had one wish*
*if i saw you*
*if i saw you*
*if i had one wish*

Another poet, human this time, who has written in this rhythmic style is the German surrealist Kurtz Schwitters:

*What a b what a b what a beauty*
*What a b what a b what a a*
*What a beauty beauty be*
*What a beauty beauty be*
*What a beauty beauty beauty be be be*
*What a be what a b what a beauty*

[1]Plymouth University, PL4 8AA, UK. Email:{alexis.kirke, eduardo.miranda}@plymouth.ac.uk.

*What a b what a b what a a*
*What a be be be be be*
*What a be be be be be*
*What a be be be be be be be a beauty be be be*
*What a beauty.*

Here is an excerpt from another of Schwitters' texts:

> *My corpse is too large, in the night - crumbles, crumbles, crumbles - too large is my corpse. Waters whip unsoftened valley - crumbles, crumbles, crumbles - too large is my corpse, giants arch dome into crumbs - crumbles, crumbles, crumbles, my corpse is too large, Cagliostro's shroud - crumbles crumbles crumbles - my corpse is too large, for the orphanage alms-for-the-poor - crumbles crumbles crumbles - too large is my corpse…*

This form of poetic text is not meaningful in the normal sense, but the sound and word repetition generate meaning. Like normal English, there is actually a hierarchical structure to the repetition (i.e. repetitions within repetitions), and the words are often evocative and sometimes contrasting. The structure of MASTER leads to poetic text of this type, as will now be seen. This approach to generative poetry is consistent with issues founds in AI poetry research, where there can be a trade-off between semantic clarity and rhythmic interest [11]

## 3 'quiet': A POEM BY MASTER

*quiet  book comet and fornicate  quiet*
*tourist ignite live quiet  quiet  book comet and wine ejaculate*
*and boring  welfare fire with fornicate*
*quiet  book comet and rape  boring  fatigued sadness it quiet*
*tourist ignite live quiet  quiet  book comet and wine ejaculate and hysterical rage collaborations fornicate  quiet  tourist*
*ignite live quiet*
*quiet  book comet and wine ejaculate and boring*
*welfare fire with hysterical explosion sensations*
*explosion explosion provoked explosion*
*explosion prizes quiet  quiet*

*quiet  book comet and wine ejaculate and quiet  quiet  book comet and fornicate*
*quiet  tourist ignite live quiet*
*quiet  book comet and wine ejaculate and boring  welfare*
*fire with fornicate  quiet  book comet*
*and rape  boring  fatigued sadness*
*it quiet  tourist*
*ignite live quiet*
*quiet  book comet and wine ejaculate*
*and hysterical rage collaborations fornicate*
*quiet  tourist ignite live quiet  quiet  book comet and wine ejaculate and boring*
*welfare fire with hysterical explosion sensations explosion explosion provoked*
*explosion explosion prizes want explosion and huge explosion and*

This poem was written as a "collaboration" between the first author of this paper and MASTER. The author provided the title and line breaks. MASTER produced the text. This particular implementation of MASTER involved 8 agents who had a slight "depressive" tendency, and 3 of whom were initially "happy", 3 initially "relaxed" and 1 "angry" and 1 "sad". The poem comes from agent 8, after 16 interaction cycles. These terms, and the MASTER system, will now be explained in more detail.

## 4 MASTER

Before introducing the emotional intelligence of MASTER, affective representation will be briefly discussed. The dimensional approach to specifying emotion utilizes an n-dimensional space made up of emotion "factors". Any emotion can be plotted as some combination of these factors. For example, in many emotional music systems [12] two dimensions are used: Valence and Arousal. In that model, emotions are plotted on a graph (see Figure 2) with the first dimension being how positive or negative the emotion is (Valence), and the second dimension being how intense the physical arousal of the emotion is (Arousal). For example "Happy" is high valence high arousal affective state, and "Stressed" is low valence high arousal state.

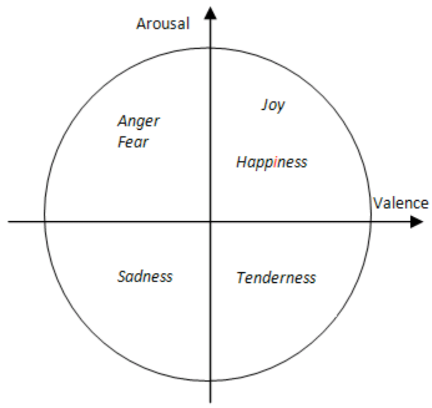


**Figure 2**: The Valence/Arousal Model of Emotion

Part of the core "emotional intelligence" in the agents in MASTER comes from a 1000 word database called ANEW [13] (Affective Norms for English Words) which each agent has internalized. This pre-prepared database contains a list of words which have had their valence and arousal measured by extensive human experiments. Each human subject was presented with single words and asked to represent their emotional response in a simple computer-based graphical system. The compiled and averaged results have been made available as a database online for academic work, and it is these that are used here. For example in the database "ace" has an average valence of 6.88 and "accuse" has an average valence of 2.54 – i.e. rated significantly less emotionally positive. Similarly "alert" has an average arousal of 6.85 whereas "affection" has a much lower average arousal of 0.86. In the current version of MASTER all agents have the same 1000 words database (although there is nothing to prevent a user from allowing agents to have different emotional word databases.)

Poems are written in MASTER by allowing the agents interact in a specific way. This interaction cycle is shown in Figure 3. In the next two sections, the modules in the diagram will be examined.

## 5 AGENT A RECITES ITS TEXT

An agent starts with an initial emotional state. This can be neutral (e.g. valence and arousal set to 0), or some bias (e.g. "depressed" with valence = -1, "excited" with arousal = 1, etc). An agent will also have an Initial Text. This can be a single word chosen by the user, or selected from a database. Agents then take it in turn to recite their text. They will recite to every other agent. This is called a single Cycle. Then it is a second agent's turn to recite for a Cycle, and so forth. So if there are 4 agents it takes 4 interaction cycles for them all to have recited their text to each other.

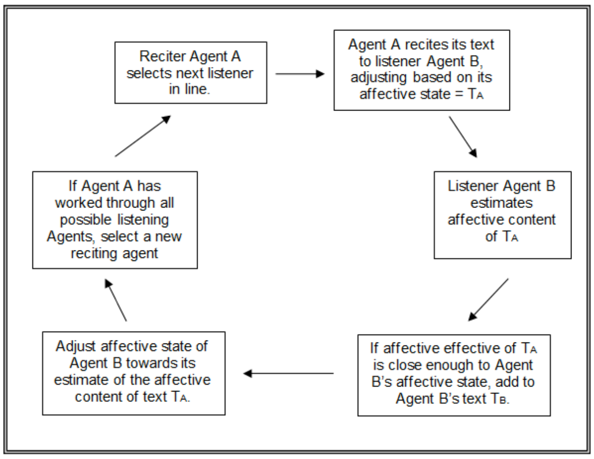


**Figure 3.** MASTER interaction cycle

An agent's recitation is adjusted by its emotional state. Firstly the reciting agent estimates the valence and arousal of its own stored internal text/poem. To measure valence the agent locates the valence values for all words in its poem which are included in its Emotional Text Database. It calculates the average of these. Not all words in the agent's text will be in the Emotion Database, so it ignores these in the averaging. For example, suppose an agent has the text "Happy smelly death". "Smelly" is not in the ANEW database so will be ignore. But "happy" has valence 0.82 and "death" has valence -0.64. The valence of the phrase is thus calculated as 0.09 (the mean of happy and death).

Arousal is calculated slightly differently. As well as calculating the average arousal from the database, average word length is used. There have been studies that examine the parallels between music and speech [14], in particular ones that support that we understand emotions expressed through music because the music mimics the way emotions are expressed in speech [15, 16]. Because music that has a higher tempo generally expresses a higher arousal [17], MASTER utilizes the concept that phrases with longer words represent a high speech tempo, and thus a higher arousal. Texts with longer (many-syllable) words will tend to read more rapidly, whereas texts with shorter words will tend to have more intra-word gaps and be read more slowly. So in MASTER the longer the average word length, the lower the calculated arousal. The formula is shown in Equation 1. The precise weightings in the formula are designed to combine with the types of values found in the ANEW database, and also to lead to total arousal values of the order -1 to 1 where possible (as commonly used in many valence / arousal models).

$$arousalEstText = 2*average(wordLength)/3 - 1 \qquad (1)$$

For example: "Happy Smelly Death" will have a higher arousal (0.87) than "Happy as Death" (0.57), because its average word length is greater.

When estimating the arousal of its internal text, an agent also uses its ANEW database. Then it combines the value in the ANEW database (if the word is in the database) with the value calculated in equation (1), as shown in equation (2), weighting the database arousal contribution twice as much as the word length calculation.

$$arousalEst = (2/3)*arousalDatabase + (1/3)*arousalEstText \qquad (2)$$

Part of the logic behind this weighting is that the ANEW database is a highly tested approach to word emotion, whereas equation (1) is very much heuristic and has not been tested on human subjects.

Note that for a human listener the actual affective impact of a word in a sentence is dependent on the words around it – i.e. its context. As a result many systems developed to analyze text emotionally incorporate this context, for example [18]. MASTER's usage of a model where valence and arousal are largely based on individual words' valence and arousal is thus an approximation, but one judged sufficient for this first implementation, and particularly for the type of poetry being examined.

At the end of the above estimation process agent A will then have an estimate of the affective content of its internal stored text. Once an agent has estimated the emotional content of its stored text, it compares this to its own emotional state (its own valence and arousal). If its valence is different to its internal text, the agent adds an emotional word to the end of its text when reciting it to another agent. This is to raise or lower the valence of the text to bring it in line with how it's feeling. It does this by searching through the database for a word whose valence will pull the phrase's estimated valence up or down towards the agent's own current valence, whilst keeping arousal roughly the same.

Similarly if the agent's current arousal is different to its estimation of the arousal of its internal text, then while reciting the agent adds to its text. Firstly, as with valence, it searches the emotion database for a word which will help to adjust its text arousal (but not its valence), and it adds the word to the end of its text. Secondly, it attempts to change the average word length of its phrase while reciting it. This is done using another database the agent has. This is a database of "neutral" words – the Neutral Database. In the current version of MASTER each agent has the same neutral database, provided by the user.

The neutral database can be generated by compiling text from source material from online (e.g. poems, articles etc.) This text is then searched and words are in the emotion database are removed. The remaining list of words is used as the neutral database. This process allows for the user to adjust the neutral database to change the nature of the final generated poems. For example, a neutral database of one of the keynote poets at the Poetry and Source Conference 2012, Plymouth, UK was used in the creation of "quiet".

So the agent searches for a neutral word of an appropriate length to change its text arousal in the right way. For example if it wants to increase the text arousal, it searches for a longer neutral word. If it wants to decrease text arousal it searches for a shorter neutral word. This is based on equation (1).

**5.1 An Example Recital**

Suppose agent A has the text "Happy smelly death" and currently has a low valence of -0.5 and high arousal of 0.5 (e.g. "angry"). The ANEW database will estimates the valence of this text as 0.09. So because the agent is "feeling" pretty negative, it wants to adjust the valence of the text to be more negative. It could adjust the sentence valence downwards (from 0.09 towards -0.5) by adding the word "bad" (valence -0.43 from the ANEW database) to its recited text. (Note – these changes are only applied to what the agent recites, not to the text that is stored.)

Now it estimates the arousal of its phrase as 1.06, using the ANEW database and equation (2). It could adjust recital arousal downwards (from 1.06 towards 0.5) by adding the word "calm" (arousal 0.42 in the database) to its recited text. To try and reduce arousal further agent A adds neutral words It adds "of" which has an estimated arousal of 0.33 using equation (1).

So the final text becomes: "Happy smelly death bad calm of". The first three words are Agent A's internal text. The next is to reduce valence, and the last two are to reduce arousal. Using ANEW and equation (2), the calculated valence and arousal of this recital are: valence of 0.06 and arousal of 0.67. So according to that, the agent has decreased valence too little, and increased arousal too little as a result of trying to match its own valence and arousal values (-0.5 and 0.5).

## 6 AGENT B ESTIMATES AFFECT AND ADDS

When Agent B hears agent A's recitation it estimates the affective content in the same way that that Agent A estimated it in Section 5 above. In other words it estimates the valence from the database, and the arousal using the database and equation (2). Then Agent B will compare that value to its own arousal and valence. If they are close enough in value then Agent B adds Agent A's text to the end of its own internal text – thus updating its internal text. So for example if Agent A recites something "happy" and Agent B feels "happy", then Agent B would add the text to its own. But it Agent B was feeling "sad" it would not. It is through this addition process amongst multiple agents that poems are built up.

Whether or not Agent B adds Agent A's recital to its own text Agent B is effected "emotionally" by hearing the recital. B's own valence and arousal are moved towards the estimated valence and arousal of the recited text from A, using equations (3) and (4). This can be compared to a happy person hearing a sad poem from another person, and it depressing them slightly.

$$valence'_B = (1-\gamma_v)valence_B + \gamma_v valenceEst_A \qquad (3)$$

$$arousal'_B = (1-\gamma_a)arousal_B + \gamma_a arousalEst_A \qquad (4)$$

Thus the interaction of the Agents could be summarized as follows. Agents recite texts to each other, adjusting the recital (adding words to it) based on their emotional state. The agents influence each other's emotional state by the text they recite. When an agent hears a text which has an emotion content close to the way it is "feeling", it adds that text to the end of its own. Thus MASTER is a society of emotional agents who generate in parallel a collection of ever growing poems based on trying to influence each other's emotional states (and communicate their own.)

It could be asked: why don't agents simply adapt their emotional state directly based on other agents' emotional state, rather than via recited words? One reason is that this indirect emotional adaption creates a more interesting dynamical system to generate unpredictable behaviour for creative reasons. A second – less significant - reason for the indirect design is that in fact humans cannot adapt to each other's emotions. We can only estimate other's emotions based on external factors we experience – such as a person's physical behaviour, or tone of voice. We cannot read minds. Thus the situation with MASTER mirrors the human social situation.

There is however a key way in which MASTER differs from emotional influence in humans. When a human aims to influence the emotions of another, they take into account the state of the other human's emotions. So if a happy person wishes to make an angry person happy, they may say different things compared to if they wish to make a depressed person happy. This is the approach that is utilized in much emotional modelling for developing agents in affective computing applications [19, 20]. However for the purposes of a simplified initial implementation, this element is currently not incorporated into MASTER. An agent in MASTER only adjusts its recital based on its own affective state, not the affective state of the agent it is reciting to.

## 7 ANALYZED EXAMPLE

Consider a MASTER example made up of 3 Agents, with initial valence / arousal states of -0.5/0.5, 0.5/0.5 and 0.5/-0.5. Anthropomorphically these could be thought of as angry, happy and relaxed. The Affective Similarity Threshold is how close an agents affective state must be to the recited text it hears before adding it to its own. This is calculated as the Euclidean distance in the valence / arousal space, and is set to 0.55 for this example. Agents are initialized each with a single word - the word in ANEW whose emotional state is closest to their own emotional state. For example for Agent 2 it is a Happy word (since its initial valence is 0.5, arousal 0.5). So its initial word is "Conquer". A value of 0.1 for the gamma sensitivities in (3) and (4) used. The 3 agents are then left to interact for 20 cycles, with the results shown below:

**Cycle 1:**

Agent 1's ("angry" agent) initial Text: *"hostage"*
Agent 2's ("happy" agent) initial Text: *"conquer"*
Agent 3's ("relaxed" agent) initial Text: "relaxed"

**Cycle 10:**

Agent 1 (slightly more happy): *"hostage conquer relaxed bird marry a erotic explosion anticonsumerist relaxed bloom extreme one"*
Agent 2 (slightly more angry): *"conquer relaxed bird marry a relaxed soothe extreme at"*
Agent 3 (slightly more happy): *"relaxed"*

**Cycle 15:**

Agent 1 (more happy again): *"hostage conquer relaxed bird marry a erotic explosion anticonsumerist relaxed bloom extreme one conquer relaxed bird marry a relaxed soothe extreme at hostage conquer relaxed bird marry a erotic*

*explosion anticonsumerist relaxed bloom extreme one extreme shock this infatuation explosion slide"*
Agent 2(more angry again): *"conquer relaxed bird marry a relaxed soothe extreme at hostage conquer relaxed bird marry a erotic explosion anticonsumerist relaxed bloom extreme one extreme shock this"*
Agent 3(more happy again): *"relaxed conquer relaxed bird marry a relaxed soothe extreme at hostage conquer relaxed bird marry a erotic explosion anticonsumerist relaxed bloom extreme one extreme shock this infatuation explosion slide"*

The system can be examined more deeply by looking in detail at the emotional internals of a single agent, Agent 1. These changes are shown in Figure 4. Agent 1 starts "Angry", then gradually arousal and valence increases because of influence of the recitals from the happy and relaxed agents. So the agent gets "happier". The agent's internal text estimate approximately tracks this change in emotion, perhaps because of the affective threshold. Agent 2 and 3's emotion evolution is shown in Figure 5.

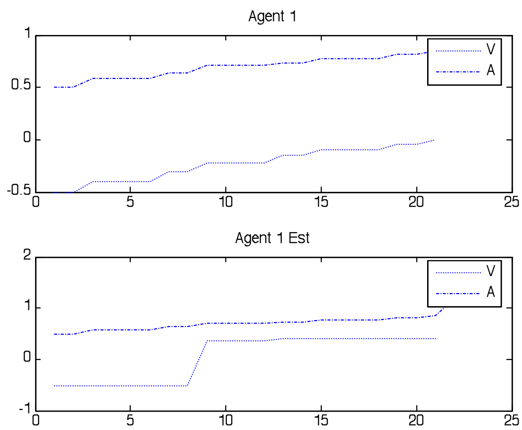


**Figure 4**: Emotional evolution of Agent 1: Internal state (Top graph); Internal Text Affective Estimate (Bottom Graph)

Simply changing the initial words will change the evolution. For example – requiring that the arousal and valence of the first three initialising words be more emotionally positive and or higher arousal (in this case increased them by 0.4) makes the words selecting from ANEW come up as: "shock", "orgasm" and "snuggle". Then Agent 1's text at 15 cycles becomes:
*"shock orgasm shock chaos rage it anxious extreme and orgasm shock chaos rage it snuggle pillow power a shock orgasm shock chaos rage it anxious extreme and hysterical rage the hysterical extreme this snuggle free explosion on orgasm shock chaos rage it snuggle pillow power a shock orgasm shock chaos rage it anxious extreme and hysterical rage the snuggle home extreme at shock orgasm shock chaos rage it anxious extreme and orgasm shock chaos rage it snuggle pillow power a shock orgasm shock chaos rage it anxious extreme and hysterical rage the hysterical extreme this snuggle free explosion on explosion shark this explosion extreme this"*

## 8 QUIET

The poem "quiet" used as the introductory example in Section 3 came from an 8 agent system. The makeup of the initial population was 3 happy and 3 relaxed agents, 1 angry and 1 sad agent. The initial word selected was much lower in valence than arousal than the agent was "feeling" – it was required to be 0.4 below each agent's arousal and valence. The Affective Similarity Threshold was set to 0.43. The poem is the internal text of Agent 8 after 16 cycles. Agent 8's emotional evolution as it wrote the text for quiet can be seen in Figure 6.

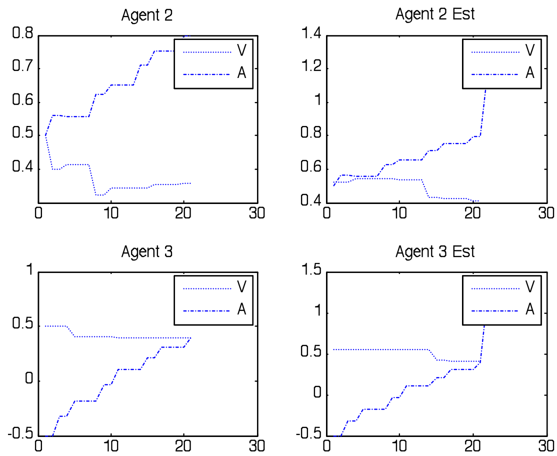


**Figure 5**: Emotional evolution of Agents 2 and 3

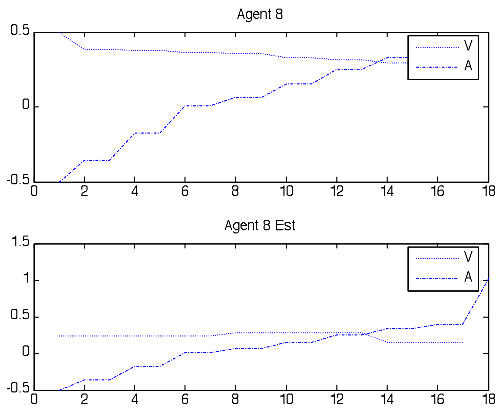


**Figure 6**: Emotional evolution of Agent 8 as it wrote "quiet"

## 9 CONCLUSIONS AND FUTURE WORK

MASTER is the first multi-agent system approach for computer-aided creation of poetry and, as far as we are aware, the first generative poetry system utilizing artificial emotion. This combination of social interactions and emotional dynamics allows the system to avoid all random processes, which are often required by creative systems [21]. The creativity emerges as a result of the complex interactional dynamics. The resulting texts are not meaningful in the normal sense - the sound and word repetition generates meaning. Like normal English, there is actually a hierarchical structure to the repetition (i.e. repetitions within repetitions), and the words are often evocative and sometimes contrasting.

There are a number of key areas which would benefit from further work. One is the emotional estimation system. In particular the arousal detection system utilizes ideas which need to be more fully tested, perhaps by perceptual studies. Furthermore the emotional estimation system is only on a word level. Emotion is generated by text through the cumulative effect of many words, phrases, stanzas and so forth. MASTER has no embedded sense of this process. For example a phrase made up of 3 happy words and 3 sad words is not necessarily emotionally neutral. In fact in many cases there may be little correlation between the emotive effect of a stanza and the emotive effect of its individual words.

Even though MASTER is not designed to write sentences, it would benefit from a clearer "understanding" of language structure. Emotional impact may be increased if the orderings of words in a MASTER text are a little more reminiscent of normal writing. Or at the very least it would be useful tool to have a parameter that allowed this to be implemented. Such a method could involve simple statistical models of word orderings, and an agent only adding text to the end of its own text if there is a sufficient statistic likelihood of such word orders.

Despite these limitations it is hoped that MASTER indicates the potential for the use of affective computing in generative poetry, and additionally indicates the potential of multi-agent systems in this field.

## REFERENCES


1. Miranda, E.R. Special Issue: Aesthetic Decisions in Computer-Aided Composition, *Contemporary Music Review*. 28(2), Taylor and Francis (2009)
2. Kirke, A. and Miranda, E. R. MASTER: Towards the use of emotional multi-agent systems in algorithmic writing and poetry. *Presentation, Contemporary Poetry and Source 2012*, Plymouth University, Plymouth UK (2012).
3. Dahl, V., Jiménez-López, M.D. and Perriquet, O. Poetic RNA: Adapting RNA Design Methods to the Analysis of Poetry, *Trends in Practical Applications of Agents and Multiagent Systems, Advances in Intelligent and Soft Computing*, 71:403-410 (2010)
4. Riedl, M.O. and Young, R.M. An Intent-Driven Planner for Multi-Agent Story Generation. *Proceedings of 2004 Conference on Autonomous Agents and Multi-agent Systems*, July 19-23, New York, New York, USA, ACM (2004)
5. Nairat, M., Dahlstedt, P. and Nordahl, M.G. Character evolution approach to generative storytelling, *Proceedings of 2011 IEEE Congress on Evolutionary Computation*, pp.1258-1263 (2011)
6. Wooldridge, M.. *An Introduction to Multi-Agent Systems*. John Wiley and Sons (2004)
7. Bentley, P.J. Everything Computes, *Proceedings of the Third International Conference on Generative Art*, Melbourne, Australia (2005)
8. Gervas, P. An expert system for the composition of formal spanish poetry. *Journal of Knowledge-Based Systems* 14(3–4):181–188 (2001)
9. Kurzweil, R., Ray Kurzweil's Cybernetic Poet, *Kurzweil CyberArt Technologies* (2000)
10. Toivanen, J.M., Toivonen, H., Valitutti, A. and Gross, O. Corpus-Based Generation of Content and Form in Poetry, *Proceedings of 2012 International Conference on Computational Creativity*, Dublin, Ireland (2012)
11. Manurunga, R., Ritchie, G. and Thompson, H., Using genetic algorithms to create meaningful poetic text, *Journal of Experimental and Theoretical Artificial Intelligence*, 24(1), Taylor and Francis (2012)
12. Kirke, A. Application of Intermediate Multi-Agent Systems to Integrated Algorithmic Composition and Expressive Performance of Music, PhD Thesis, Plymouth University, Plymouth, UK (2011)
13. Bradley, M., Lang, P. Affective norms for English words (ANEW): Stimuli, instruction manual and affective ratings. *Technical report C-2*, Gainesville, F. The Center for Research in Psychophysiology, University of Florida, USA (2010)
14. Jackendoff, R. Parallels and Non-Parallels between Language and Music. *Music Perception*, 26:95-204 University of California Press (2009)
15. Bowling, D.L., Sundararajan, J., Han, S., Purves, D. Expression of Emotion in Eastern and Western Music Mirrors Vocalization. *PLoS ONE* 7(3): e31942. doi:10.1371 / journal.pone.0031942 (2012)
16. Francis, N. Music contact and language contact: A proposal for comparative research, *The Linguistic Review*. 30(1):1–24, Walter de Grutyer (2013)
17. Kirke, A. and Miranda, E. R. Combining EEG Frontal Asymmetry Studies with Affective Algorithmic Composition and Expressive Performance Models. *Proceedings of 2011 International Computer Music Conference*, Huddersfield, UK (2011)
18. Strapparava, C., Mihalcea, R. Learning to identify emotions in text. *Proceedings of the 2008 ACM symposium on Applied Computing*, pp.1556-1560, ACM New York, NY, USA (2008)
19. Asteriadis, S., Tzouveli, P., Karpouzis, K., Kollias, S. Estimation of behavioral user state based on eye gaze and head pose—application in an e-learning environment, *Multimedia Tools and Applications*, 41(3):469-493, Springer (2009)
20. Tao, J., Tieniu, T. Affective Computing: A Review. Affective Computing and Intelligent Interaction. *LNCS* 3784, pp. 981–995, Springer (2005)
21. Montfort, N. and Fedorova, N. Small-Scale Systems and Computational Creativity. *Proceedings of 2012 International Conference on Computational Creativity*, Dublin, Ireland (2012)