

Adapting a Generic Platform for Poetry Generation to Produce Spanish Poems

Hugo Gonalo Oliveira CISUC, Dep. Engenharia Informtica
Universidade de Coimbra Portugal
hroliv@dei.uc.pt

Raquel Hervs and Alberto Daz D. de Ing. de Software e Int. Artificial
Universidad Complutense de Madrid Spain
{raquelhb, albertodiaz}@fdi.ucm.es

Pablo Gervs Inst. de Tecnologa del Conocimiento
Universidad Complutense de Madrid Spain
pgervas@sip.ucm.es

Abstract

PoeTryMe was created as a generic system for the generation of poetry that takes into account both semantics, in the form of triplets of relations between concepts, and textual structure, in the form of a grammar of templates extracted from existing poems. It was originally instantiated to generate poetry in Portuguese. The present paper describes an effort to create a different instantiation of PoeTryMe, this time focused on the production of poems in Spanish. The instantiation effort involved the creation of a set of triplets of relations to represent the semantics of Spanish terms, the extraction of a grammar of templates for Spanish from a corpus of Spanish poetry, the application of a different tool for Spanish syllabic division, the integration of the various modules, and several experiments with the resulting system.

Introduction

Existing efforts at the automatic generation of poetry in recent years have uncovered a number of methods for implementing computationally this task, both from a semantic seed (Manurung 2003; Manurung, Ritchie, and Thompson 2012) and from a set of templates (Oulipo 1981; Colton, Goodwin, and Veale 2012), or by combining both (Gonalo Oliveira 2012; Veale 2013). Yet most existing efforts consist of custom-tailored solutions for specific languages, and it is difficult to envisage what amount of effort might be required to port one of them from the language for which it was originally designed to a different language. The present paper addresses the question of exploring the effort required by the task of adapting an existing generic platform for poetry generation, PoeTryMe (Gonalo Oliveira 2012), to a language (Spanish) different from the one over which its original instantiation was designed (Portuguese). To produce its poems, the PoeTryMe platform combines both semantic information, in the form of relation triplets between concepts which are used during the selection of content for the poems, and textual information, in the shape of template-like grammar rules used to render as text the selected content. In this sense, it presents a special challenge because resources specific to the new target language need to be produced at both levels, semantics and textual.

The present paper reports on the engineering and development effort for these required resources and presents an

exploration of the effect of their characteristics on the performance of the poem generation process. Throughout this effort, the overall goal has been to reuse or adapt existing resources and/or to extract automatically any novel ones, in order to avoid as much as possible the risks of *fine tuning* (Colton, Pease, and Ritchie 2001) inherent in hand-crafting them.

Previous Work

Over recent years, many efforts that address the study of creativity from a computational point of view acknowledge the work of Margaret Boden (Boden 1990) as a predecessor. 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. Poetry generation systems explore a conceptual space characterised by form and content. The concept of *articulation* (Gervs 2013) describes the initial analysis of a target artifact with a view to select a particular frame for understanding and decomposing it into parts that can later be used to assemble equivalent instantiations of the same type. This captures both the concept of different parts being joined together in a whole and the concept of allowing the parts to move with respect to one another. Different decisions on articulation can lead to processes that select a particular textual template with which the poems are produced (Oulipo 1981; Colton, Goodwin, and Veale 2012), or reuse a predetermined set of verses (Queneau 1961), or draw upon given sets of lexical items to employ (Gervs 2001), or even rely on a language model to follow, obtained from a reference corpus (Barbieri et al. 2012). The degree of articulation determines a particular conceptual space of possible poems, with poems outside that space being unreachable unless the articulation is changed.

In terms of computational techniques used to explore these conceptual spaces, several solutions have been applied. The generate & test paradigm of problem solving has also been widely applied in poetry generators such as the early version of the WASP system (Gervs 2000b) and the initial work by Manurung (Manurung 1999). Evolutionary solutions have as well been applied (Manurung, Ritchie, and Thompson 2012). An evolution of the WASP system (Gervs 2001) used case-based reasoning (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. Alternative approaches to poetry generation include the application of constraint programming techniques (Toivanen, Järvisalo, and Toivonen 2013), which has a great potential for adequately modelling the large amount of constraints that poetry generation deals with.

Although a pleasant sound and a regular rhythm can sometimes make up for poor or inexistent semantics (Gonçalo Oliveira, Cardoso, and Pereira 2007), meaning is also seen as an important feature in computer-generated poetry, whether more precise, vague or figurative. (Veale 2013) describes a system heavily influenced by semantic information, used to drive the poetry generation process, with special focus on figurative language and rhetorical tropes. But different systems handle semantics differently. In evolutionary approaches, among the other constraints, the goal state should consider meaning (Manurung 2003; Manurung, Ritchie, and Thompson 2012), whereas in CBR approaches, words are selected according to a given prose message. In fact, in several systems generation starts with a theme or a set of seed words, which constrain the poem search space and may be seen as setting the semantics of the poem (Wong and Chun 2008; Netzer et al. 2009; Yan et al. 2013). The choice of relevant words may be achieved either with the help of semantic knowledge bases (Netzer et al. 2009; Agirrezabal et al. 2013), by exploring models of semantic similarity, extracted from corpora (Wong and Chun 2008; Toivanen, Järvisalo, and Toivonen 2013; Yan et al. 2013), or both (Colton, Goodwin, and Veale 2012).

PoeTryMe

PoeTryMe, originally presented in (Gonçalo Oliveira 2012), is a poetry generation platform, on the top of which different systems for poetry generation can be implemented. It relies on a modular architecture (see Figure 1), which enables the independent development of each module and provides a high level of customisation, depending on the needs of the system and ideas of the user. It is possible to define the semantic relation instances to be used, the sentence templates of the generation grammar, the generation strategy and the configuration of the poem. In this section, the modules, their inputs and interactions are presented.

Generation Strategies

A Generation Strategy organises sentences according to some heuristics, such that they suit, as much as possible, a target template of a poetic form and exhibit certain features. A poem template contains the poem's structure, including the number of stanzas, lines per stanza and of syllables in each line. Templates may also use a symbol for denoting the target rhyme for the lines. Figure 2 shows poem structure templates for a haiku (5-7-5) and a sonnet (14*10-syllable verses). There is no rhyme pattern specified for the haiku, but each line of the sonnet has a symbol that results in the following rhyme pattern: *ABBA ABBA CDC DCD*.

Each strategy uses the Sentence Generator module to retrieve natural language sentences, which might be selected as poem lines. For the generation of a poem, a set of seed

```
#haiku
stanza{line(5);line(7);line(5)}

#sonnet
stanza{line(10:A);line(10:B);line(10:B);line(10:A)}
stanza{line(10:A);line(10:B);line(10:B);line(10:A)}
stanza{line(10:C);line(10:D);line(10:C)}
stanza{line(10:D);line(10:C);line(10:D)}
```

Figure 2: Templates with the structure of a haiku and a sonnet with a rhyme pattern.

words is provided and used to narrow the set of possible generations, this way defining the generation domain.

An instantiation of the Generation Strategy does not generate sentences, but follows a plan to select the most suitable sentences for each line. Selection heuristics might consider features like metre, rhyme, coherence between lines or other, depending on the desired purposes. Some of these features are evaluated with the help of the Syllable Utils.

Syllable Utils As its name suggests, this module consists of a set of operations on syllables. Given a word, Syllable Utils may be used to divide it into syllables, to find the stress, or to extract its termination, useful to identify rhymes.

Sentence Generator

This is the core module of PoeTryMe. It is used to generate meaningful natural language sentences, with the help of:

- A semantic graph, managed by the Relations Manager, that connects words according to relation predicates (see Figure 3 for a very simple semantic graph, centered in the word *poetry*, in Portuguese/English).
- Generation grammars, processed by the Grammar Processor, with textual renderings for the generation of grammatical sentences that express semantic relations.

The generation of a sentence starts by selecting a random relation instance, in the form of a *triplet* = {*word*₁, *predicate*, *word*₂}, from the semantic graph. Then, a random rendering for the predicate of the *triplet* is retrieved from the grammar. After inserting the arguments of the *triplet* in the rule body, the resulting sentence is returned. A third module, the Contextualizer, keeps track of the instances that were used to generate the lines and may be used to explain the choices made.

Relations Manager The Relations Manager is an interface to the semantic graph. It may be used to retrieve all words related to another, or to check if two words are related by indicating their relation.

To narrow the space of possible generations, a set of seed words is provided to the Relations Manager. This set defines the generation domain represented by a subgraph of the main semantic graph, where the relation triplets should either contain one of the seed words or somehow related words. More precisely, the subgraph will only contain triplets with words that are at most δ nodes far from a seed word, where δ is a

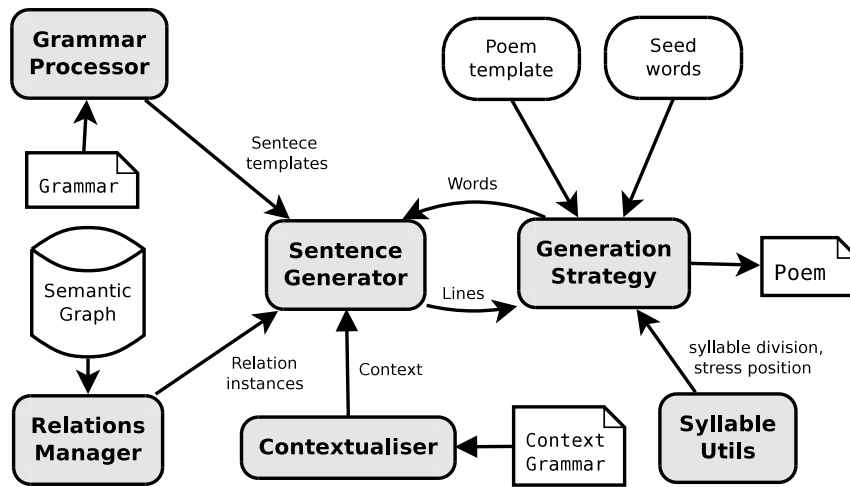


Figure 1: PoeTryMe's architecture

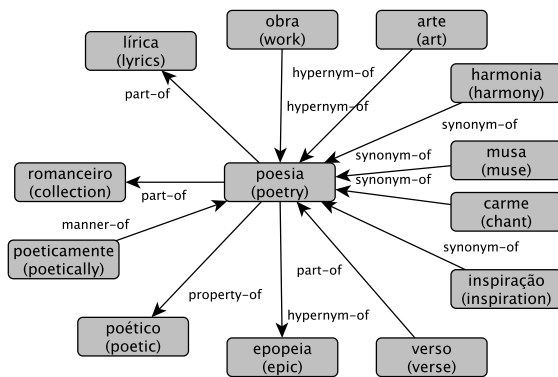


Figure 3: Semantic Graph example

neighbourhood depth threshold. It is also possible to define a surprise factor, ν , interpreted as the probability of selecting triplets one level further than δ .

The number of seed words is open, and it can be enlarged with the top n relevant words for those seeds. For this purpose, the PageRank (Brin and Page 1998) algorithm is run in the full semantic graph. Initial node weights are randomly distributed across the seeds, while the rest of the nodes have an initial weight of 0. After 30 iterations, nodes will be ranked according to their structural relevance to the seeds. The n higher ranked nodes are selected.

Grammar Processor The Grammar Processor is an interface for the generation grammar. Similarly to Manurung (Manurung 1999), it performs chart generation with a chart-parser in the opposite direction. A grammar is an editable text file with a list of rules, whose body should consist of natural language renderings of semantic relations and there must be a direct mapping between the relation names, in the graph, and the rules' name, in the grammar. Besides simple terminal tokens, that will be present in the poem

without change, this module supports terminal tokens that indicate the position of the relation arguments ($\langle arg1 \rangle$ and $\langle arg2 \rangle$), to be filled by the Sentence Generator. This way, given a relation predicate, the Grammar Processor can retrieve one (or several) renderings for any triplet of that kind.

A very simple example of a valid rule set, with three hypernymy patterns, is shown in Figure 4. These rules could be used to generate sentences as: *a tool like a hammer, mango is a delicious fruit, man before animal*.

```

HYPERNYM-OF → a <arg1> like a <arg2>
HYPERNYM-OF → <arg2> is a delicious <arg1>
HYPERNYM-OF → <arg2> before <arg1>

```

Figure 4: Grammar example rule set.

Contextualizer The ability to explain how its artefacts are created is an important feature of a creative system. PoeTryMe provides this feature by keeping track of all the relation instances that originated each line. Towards the notion of framing (Charnley, Pease, and Colton 2012), these can later be used to contextualize the poem by indicating the relation instances used to form the lines and how they are connected to a word in the generation domain. The context can be a mere list of relation instances or, if a contextualisation grammar is provided, it may consist of a natural language piece of text.

Generating Poetry in Spanish

The process of instantiating the PoeTryMe platform to generate Spanish poems required three separate processes of relevant system resources: (i) construction/adaptation of Spanish lexical resources (morphological lexicon, lexical-semantic knowledge base, syllable division tool); (ii) construction/extraction of a set of template-like renderings; and

(iii) configuration of an appropriate generation strategy. Before describing those processes, some remarks on the requirements and on the flexibility of PoeTryMe are provided.

Remarks on Requirements and Flexibility

As presented in the previous section, PoeTryMe’s architecture is very flexible and may be used to generate poetry in different languages and/or on different domains. This applies as long as there are three main tools available, namely a lexical-semantic network, a generation grammar and syllable utilities, all targeting the same language.

The lexical-semantic network, handled as a semantic graph, can be broad-coverage or on any specific domain, as long as it contains relation instances represented as triplets ($word_1$ related_to $word_2$). The generation grammar must contain textual renderings for the relation types covered by the lexical-semantic network. And the syllable tool should at least provide a method for each of the following operations: splitting a word into syllables, stress identification and termination extraction.

As a lexical-semantic network typically contains only lemmatised words, if we want to use also inflected words, a morphological lexicon might also be needed in a pre-processing step. This lexicon should be as broad as possible and provide the part-of-speech (POS) of the words of the target language, as well as other morphological information, such as the gender and number of nouns and adjectives. It can be used for adding inflected words to the lexical-semantic network and contribute to more variation. Moreover, if the generation grammar is learned automatically, with the help of the network, it will enable to learn more complete grammars.

For Portuguese, there have been different instances of PoeTryMe where, apart from different generation strategies, the main differences in external resources were the different sizes of the lexical-semantic network and of the generation grammar. In fact, in the first instantiations of PoeTryMe, the generation grammars were handcrafted. Regarding the adaptation to Spanish, we used a morphological lexicon with the same information as the Portuguese, a syllable tool that performed the same operations, and a lexical-semantic network with the same format. The main difference probably relies on the latter. While, for Portuguese, the lexical-semantic network was extracted automatically from dictionaries (CARTÃO (Gonçalo Oliveira et al. 2011)), for Spanish, it was obtained from a handcrafted resource. This resulted in a larger semantic graph for Portuguese (about 286,000 triplets between lemmas) covering more relation types, and more figurative language, but also more imprecisions. Another obvious difference on the instantiations for different languages results from the different generation grammars, which are learned from different collections of text, each written in its own language.

Lexical Resources Used

In order to handle the inflection of nouns and adjectives (number and gender), the dictionary from FreeLing (Padró and Stanilovsky 2012) has been used as lexicon of Spanish. It contains over 650,000 inflected word forms including

nouns, verbs, adjectives and adverbs. For each form, there is information on the lemma, the POS, and inflection details that include the tense of the verbs and the number and gender of the nouns and adjectives.

As the source of relation instances that would build our semantic graph, we have used the Spanish WordNet from the Multilingual Central Repository (MCR) version 3.0 (Gonzalez-Agirre, Laparra, and Rigau 2012). MCR follows the classic wordnet structure, and thus contains synsets and relations between them. The following example shows how a synset relation is converted to relation triplets between words:

Synset relation	{ <i>automóvil, carro, coche</i> } hypernym-of { <i>coche_deportivo, deportivo</i> }
Word triplets	<i>automóvil</i> hypernym-of <i>coche_deportivo</i> <i>automóvil</i> hypernym-of <i>deportivo</i> <i>carro</i> hypernym-of <i>coche_deportivo</i> <i>carro</i> hypernym-of <i>deportivo</i> <i>coche</i> hypernym-of <i>coche_deportivo</i> <i>coche</i> hypernym-of <i>deportivo</i>

A total of 366,125 relation triplets were obtained from the MCR relation tables. Additionally, 58,052 synonymy instances were obtained from the synsets. But we did not use relations of some types, namely those indicating that some word is in a synset gloss (*rgloss*), nor those that reference a previous version of WordNet (*see_also_wn15*). After filtering, we had about 103,000 triplets, held between lemmas, to which we add all possible inflections of nouns and adjectives. In the end, this resulted in 231,296 relation triplets.

To compute the metric scansion of the poems in Spanish in terms of syllables, the corresponding module of the WASP generator of Spanish poetry (Gervás 2000b) was employed. This module is a Java reimplementación of an original set of rules designed as a logic program (Gervás 2000a). For integrating this module in PoeTryMe, an interface with the operations needed by the Syllable Utils module, and shared by the Portuguese tool, was implemented.

Learning Renderings for Semantic Relations

While we could have handcrafted generation grammars with semantic relation renderings, we decided to learn those automatically. This way, a larger and broader set of renderings was obtained, with much less manual labour.

For this purpose, we exploited a collection of human-written Spanish poetry, with poems from an existing anthology of Spanish poetry on the web¹ and also from the WASP knowledge base. Those amounted to 395 poems. The poems of this collection were processed while renderings, represented as grammars rules, were extracted from each line in the human-written poems where two words in a semantic triplet co-occurred. We used the aforementioned 231,296 triplets, collected from MCR.

Generation Strategy

In all our experiments, we have used a generate & test strategy (GT), already implemented in previous versions of

¹<http://www.poemas-del-alma.com/>

PoeTryMe. From the currently available strategies, this achieves rhymes more consistently. For each target line, GT consists of the successive generation of sentences, while keeping only the best scoring ones. Line generation stops either after a predefined number of generated sentences (n), or when a sentence is generated precisely with the target number of syllables and target rhyme, if there is one.

Sentences are first scored according to the absolute difference between their number of syllables and the target number of syllables, for the line. The higher the score, the less suited the sentence’s metre is. On the top of this score, there are bonuses for rhymes (-2 points) and penalties for sentences that end with the same word as another in the same stanza. Moreover, we may set a progressive multiplier (π) to increase the number of generations for lines of higher order in the stanzas, this way increasing the probability of rhymes.

Experimentation

Different configurations have been used to test the performance and behaviour of the system. In order to study the relation of input knowledge (both lexical and semantic) and the performance of the system, we have worked with different sets of data in the experiments.

Regarding the discovery of lexical renderings to create the final text, we have trained the system using two different sets of Spanish poems:

- The whole collection of 395 Spanish poems (GR+), which produced a total of 1,285 grammar rules.
- A subset of the previous collection with only 64 poems (GR-), which produced a total of 245 grammar rules. Note that all the grammar rules in GR- are also in GR+.

In addition, different sets of semantic relations were used:

- The whole set of semantic relations from MCR (SR+), which contains 231,296 triplets.
- A subset of SR+ with only synonymy relations (SR-Syn), which contained 55,300 triplets.
- A subset of SR+ with only hypernymy relations (SR-Hyp), which contained 130,669 triplets.

In order to produce comparable results, all the experiments were performed using the same configuration. The goal was to generate a sonnet without a predefined rhyme pattern, using the generate & test strategy (GT), with a maximum of 1000 generated sentences per line. For setting the semantic domain, two values for the neighbourhood depth threshold were tested, $\delta = 1$ and $\delta = 2$, each used to generate a set of 100 poems, always with the surprise factor $\nu = 0.1$. The seed words used were always *amor* (love), *muerte* (death), *suerte* (luck), *vivir* (to live), *sentir* (to feel), and *morir* (to die). These were chosen especially because they were the main topics in the original set of poems. PageRank was not used, so the system only worked with this exact set of seeds.

Experiments on Semantic Relations and Evaluation

Table 1 presents the results obtained regarding the semantic relations used and the evaluation scores of the result-

ing poems. The former is presented as the size of the explored subgraph, given in terms of the percentage of distinct triplets used from the full semantic graph, in each case – all (SR+), only synonymy (SR-Syn), only hypernymy (SR-Hyp). About evaluation, the presented scores gave -2 bonuses to each line ending with a termination previously used in the same stanza. As the lower the score, the better, this results in a possible best score of -20 . We recall that this is not exactly the same evaluation function used in GT. In this strategy, the best possible score for a sonnet would be -12 , because every time a rhyme occurs, the target termination is discarded. This however does not prevent the generation of poems as the one in Figure 5, where all lines share the same termination.

δ	GR	SR	% of SR	Evaluation		
				Avg.	Worst	Best
1	GR-	SR+	0.67%	-8.76	-2	-14
1	GR+	SR+	0.77%	-5.19	0	-10
2	GR-	SR+	13.80%	-8.19	-3	-13
2	GR+	SR+	17.78%	-5.93	-1	-12
1	GR-	SR-Hyp	0.56%	-10.86	-6	-19
1	GR+	SR-Hyp	0.61%	-4.68	-1	-9
2	GR-	SR-Hyp	13.04%	-12.03	-7	-19
2	GR+	SR-Hyp	15.30%	-5.53	-1	-10
1	GR-	SR-Syn	0.56%	-6.77	-3	-11
1	GR+	SR-Syn	0.55%	-4.62	0	-9
2	GR-	SR-Syn	2.49%	-8.91	-5	-14
2	GR+	SR-Syn	2.49%	-4.69	0	-10

Table 1: Use of semantic relations (SR) and evaluation results for the different configurations of the experiments

On the semantic relations used, values are consistent among different configurations. When $\delta = 2$ instead of 1, more triplets are used by definition. The increase in the percentages between $\delta = 1$ and $\delta = 2$ is proportional in all the experiments, including those using SR-Syn, where it is smaller because the full semantic graph contains about 23.9% synonymy triplets but 49.0% hypernymy.

Regarding the scores automatically assigned by the system, the average poem score is higher (and therefore less desirable) when more grammar rules are used (GR+). A possible explanation for this counterintuitive behaviour is the increased number of grammar rules without extending the cut-off values for the resulting search. It is therefore possible that the search over the larger conceptual space is cut off prematurely, thereby having less options to find exactly the combination of relations, words and renderings most appropriate from the point of view of rhyme and length. There is not a clear relation between system assigned scores and the number or type of semantic triplets used.

The best scoring poems were obtained with the smaller set of grammar rules (GR-) and only hypernymy relations (SR-Hyp). Figure 5 shows the best poem of experimental runs, along with its rough translation and the experimental configuration that lead to its production. This sonnet uses the same lexical template for all lines and adjusts it by using different pairs of verbs, where one is a hypernym of the other. The rhyme is perfect, but not especially interesting,

<i>mi hospedar no quiere albergar</i> <i>mi pensar no quiere relacionar</i> <i>mi olvidar no quiere arrojar</i> <i>mi morir no quiere soportar</i> <i>mi ocupar no quiere trabajar</i> <i>mi indicar no quiere informar</i> <i>mi recibir no quiere saludar</i> <i>mi tragarse no quiere soportar</i> <i>mi albergar no quiere albergar</i> <i>mi resolver no quiere terminar</i> <i>mi ocupar no quiere trabajar</i> <i>mi residir no quiere habitar</i> <i>mi percibir no quiere observar</i> <i>mi olvidar no quiere descartar</i>	Strategy GT	<i>my hosting wants no holding</i> <i>my thinking wants no relating</i> <i>my forgetting wants no throwing</i> <i>my dying wants no tolerating</i> <i>my busying wants no working</i> <i>my indicating wants no informing</i> <i>my receiving wants no greeting</i> <i>my swallowing wants no tolerating</i> <i>my holding wants no holding</i> <i>my resolving wants no ending</i> <i>my busying wants no working</i> <i>my residing wants no living</i> <i>my perceiving wants no observing</i> <i>my forgetting wants no discarding</i>
	Renderings, relations GR-, SR-Hyp	
	Generations/line 1000	
	$\delta + \nu$ 1.01	
	PageRank no	
	Domain amor (love) muerte (death) suerte (luck) vivir (to live) sentir (to feel) morir (to die)	
Score -19		

Figure 5: System configuration in the experiment that obtained the best-scoring sonnet

as all the lines end with ‘ar’.

On the contrary, the worst scoring poems are always obtained with the complete set of grammar rules (GR+) regardless of the semantic relations used. Figure 6 presents one of these poems where the choice of lexical templates is not as repetitive as in the best poem, but there are just no rhymes.

Besides the best and worst-scoring, from all the generated poems, we manually selected a more balanced one, which is shown in Figure 7. This choice was based on the variety of lexical templates used, metre matching, presence of rhymes, and evocative semantics.

Experiments on Grammar Rules

Table 2 has some figures on the experiments regarding the lexical renderings used from the grammar rules and the diversity on their selection. Although more configurations were tested, only those with the complete set of semantic relations (SR+) and $\delta = 1$ are shown. Results with other configurations were similar.

	Distinct renderings	Repetitions		Renderings from GR
		average	maximum	
GR-SR+	57	15.72	259	14.29%
GR+SR+	257	6.83	114	16.31%

Table 2: Use of lexical renderings for different experimental configurations

These results show that the repetition of the same rendering is very common. In both configurations, the average number of repetitions per rendering used is relatively high. The number of repetitions is even higher in the configuration with GR-. This is expected because the number of available lexical renderings is smaller and the ones suitable for the poem must be used more times.

The number of lexical renderings used from the whole set of grammar rules (GR) is quite small in both experiments. In fact, only 15% of the lexical renderings derived from the

grammar rules are used in the generated poems. This is due to the nature of the grammar rules derived from the original poems. For example, many lexical renderings correspond to lines in the original poems with significantly more or significantly less than 10 syllables. Therefore, their suitability for generating 10 syllable lines required by our sonnets is low.

In order to test the coverage of the lexical renderings in the generated poems, we carried out a process of obtaining the grammar rules implicit in the generated poems. This was done in an equivalent manner to that used for obtaining renderings from the original set of poems – the poems generated automatically were processed, and grammar rules were extracted for each line where two words in a triplet co-occurred. This led to an interesting finding: new lexical renderings, not in the original generation grammar rules, were discovered in the generated poems. From the total of lexical renderings obtained from the generated poems in both experiments (57 and 257 respectively), about 53% and 39%, respectively, were different from those in the original set of grammar rules. Considering repetitions, respectively 77% and 85% of the lexical renderings used in the poems were in the original set of grammar rules.

New renderings obtained from the generated poems are discovered because of new relations between words in the triplets and words in the final realization of grammar rules. On the one hand, the new renderings could be incorporated as new rules of the generation grammar. This would result in a broader set of more varied and possibly more interesting renderings, worth being explored in the future. On the other hand, we should take some precautions because, while the new renderings would still be grammatically correct, they might be less semantically coherent.

About the most frequent lexical rendering in all the experiments, it is “*mi <arg2> no quiere <arg1>*” (my <arg2> does not want to <arg1>) where both arguments are expected to be verbs, and <arg2> a hypernym of <arg1>. When hypernyms are not used (SR-Syn), the most frequent rendering depends on the configuration. It can be: “*quiero <arg>*”

<p><i>de vivir y poblar la fe de cristo quiero quedarse entregar el alma murió como un cabo el final quiero identificar distinguir</i></p> <p><i>murió como un gusto el afecto de poblar y vivir la fe de cristo gran muerte de matanza concurriendo quiero perder la vida sucumbir</i></p> <p><i>de vivir y durar la fe de cristo trayendo el final a fin dudoso y la desaparición y la muerte</i></p> <p><i>murió como un afecto el gusto de encontrar y dar la fe de cristo quiero percibir poner atención</i></p>	Strategy GT	<p><i>from living and populating the faith of Christ I want to stay give up my soul he died like a corporal at the end I want to identify distinguish</i></p> <p><i>he died like a pleasure the tenderness from populating and living the faith of Christ great death of slaughter concurring I want to loose my life succumb</i></p> <p><i>from living and lasting the faith of Christ bringing the ending to dubious end and the disappearance and the death</i></p> <p><i>he died like a tenderness the pleasure of finding and giving the faith of Christ I want to perceive to pay attention</i></p>
	Renderings, relations GR+, SR-Syn	
	Generations/ line 1000	
	$\delta + \nu$ 1.01	
	PageRank no	
	Domain amor (love) muerte (death) suerte (luck) vivir (to live) sentir (to feel) morir (to die)	
	Score 0	

Figure 6: System configuration in the experiment that obtained the worst-scoring sonnet

<p><i>sordos a las estimas y afectas en el dulce amor ejercitados en los presentes trabajos y cuidados hinchén de tristes desgracias el viento</i></p> <p><i>llamar oler sentir les aprovecha y cálidos indómitos cordiales por los odiosos los amables males hinchén de tristes desgracias el viento</i></p> <p><i>ocupará los actos y la pérdida hinchén de tristes desgracias el viento que ni la matanza ni el violento</i></p> <p><i>duras puentes romper cual tiernas cañas mi lamentar no quiere lamentarse mi ocupar no quiere esforzarse</i></p>	Strategy GT	<p><i>deaf to appreciations and affections in sweet love exercised in present works and cares swell the wind with disgrace</i></p> <p><i>calling, smelling, feeling profits them and warm cordial untamed by the hated the kind evils swell the wind with disgrace</i></p> <p><i>it will fill actions and loss swell the wind with disgrace that neither killing nor violent</i></p> <p><i>hard bridges to break like tender reeds my regret does not to want to regret my labor does not want to exert</i></p>
	Renderings, relations GR+, SR+	
	Generations/ line 1000	
	$\delta + \nu$ 1.01	
	PageRank no	
	Domain amor (love) muerte (death) suerte (luck) vivir (to live) sentir (to feel) morir (to die)	
	Score -7	

Figure 7: System configuration in the experiment that obtained a more balanced sonnet

<arg>” (I want <arg> <arg>), where the arguments are synonym verbs; “murió como un <arg> el <arg>” (he died as a <arg> the <arg>), where the arguments are synonym nouns; or “de <arg> y <arg> la fe de cristo” (of <arg> and <arg> the faith of Christ), where the arguments are synonym verbs.

Experiments on the Choice of Seed Words

Another set of experiments has been performed to compare the effect of using different seeds for generation.

First, the seed words used in the previous experiments were changed to study the effect of choosing seeds according to the term-frequency in the original poems. So, the six most and the six least frequent terms occurring in the original collection of poems were used. They were *yo* (I), *gente* (people), *tierra* (dust), *amor* (love), *vida* (life), and *ser* (to be). The least used terms were *abismo* (abyss), *austro* (south

wind), *tempestades* (storms), *detenerse* (to stop), *creer* (to believe), and *combatir* (to fight). These experiments were only run with the GR+SR+ configuration with $\delta = 2$. The obtained results shown a big difference regarding the number of semantic triplets used (75,622 vs 9,014), but not very different evaluation scores (on average, -5.78 vs -7).

In another experiment, instead of using a predefined set of seed words, *amor* has been chosen as initial seed and PageRank was used to obtain the top-5 most relevant words. As expected, this set contained the word *amor* itself, and four other words, including some inflections: *amores*, *cariño*, *afectas*, *afecta*. The tested configurations were GR+SR+, GR-SR+ and GR+SR-Hyp, always with $\delta = 2$. With the five previous seeds and these configurations, the best scoring poem was obtained with the complete set of grammar rules (GR+) and with the whole set of semantic relations (SR+).

Discussion

The approach followed by PoeTryMe constitutes an effort to integrate the two classic approaches to poetry generation: it combines a degree of processing to obtain the structure of the poem from a given semantic input (semantic-based generation), and resorts to a grammar of possible renderings of the semantics so obtained to provide the final syntactic form of the resulting poem (syntax-aware generation). This procedure involves a double articulation into a set of semantic elements, each coupled with one or more syntactic elements from a parallel set. This structuring of the process has a certain similarity to the work of (Manurung 1999; Manurung, Ritchie, and Thompson 2012), where logical forms taken as input semantics were paired with TAG constructions that rendered them into text.

The fact that the set of renderings is obtained from a corpus of existing poems has parallelism to a case-based reasoning approach such as the one advocated in (Gervás 2001), but the renderings themselves are closer to the templates used in the Rimbaudellaires (Oulipo 1981).

Nevertheless, the procedure has its limitations. The fact that patterns for rendering are extracted only from contexts where two terms connected by a semantic relation occur within a small distance of one another in the original set of poems is a very strong constraint. As a result only a small percentage of the total set of lines of the original poems is selected into the final set of grammar rules used for rendering. Where an articulation solution based on lines, such as the one applied by (Queneau 1961) would make every line in the original set of poems available to be included in the resulting poems, the articulation solution used for PoeTryMe restricts the conceptual space to be explored to only those lines that contain pairs of terms connected by semantic relations. This has a secondary effect in that available patterns for rendering are very unlikely to originate from lines that are contiguous in the original poems. As a result, the chances are very low for fluent connection to arise during construction between lines that follow one another in the resulting poems.

An additional restriction arises from the fact that each of the grammar rules used for rendering, by virtue of being a template with part of its constituent words already fixed during extraction, imposes a particular number of syllables that acts as starting point for the resulting line. Although different choices of words that will be employed to fill it may produce a slight variation (the final line will be longer if longer words are used, shorter otherwise), particular templates will be better suited for producing lines of length similar to that of the poem from which they originate. This could explain why such a small percentage of the extracted set of possible renderings are employed in the final set of poems, obtained with system configurations for a particular set of restrictions in terms the length of lines. Only grammar rules for renderings obtained from poems with lines of length similar to the target size are likely to be useful in producing new poems.

From the point of view of the perception of creativity that the resulting poems inspire in their readers, the first impression is surprisingly positive. Generated poems have a high degree of variation in spite of being produced by means of templates. This is due to the relative richness of different

lexical terms, achieved by the use of the Spanish wordnet. The use of semantic triplets as a constraint when filling in the templates enforces a logical connection between the various ingredients that ensures an impression of cogency. This is the result of constraints at two different levels: the existing link between each template and a particular semantic relation, and the imposition that the two terms used to fill the template be connected to one another by the corresponding relation. The metric pattern imposed by the Generation Strategy ensures that the form of the poem fulfills very closely the breakdown of lines into stanzas, the required number of syllables for each line, and, if availability of resources permits it, even appropriate rhymes.

Concluding Remarks

The present paper reports on the effort to adapt the PoeTryMe generic platform for producing poems in Spanish. This involved mostly the construction, reuse and extraction of the required resources to inform system operation. These resources were integrated with existing operational modules of the platform. The development of resources has been engineered with care to reduce the risk of fine-tuning the system towards a particular set of results. Nevertheless, the resulting set of poems shows heavy evidence of a particular style apparent in the lack of fluent grammar across sentences, a tendency to repeat successful patterns of speech (corresponding to optimal templates for lines), and a preference for infinitives as rhyming solutions.

In more general terms, the set of operational modules, strategies and configurations of input parameters available in the PoeTryMe is much larger than the limited subset that has been explored to obtain the results presented in this paper. Further work can be considered to explore the possible conceptual spaces that may be reached by applying the combinations left untried at the closure of this paper. Among other parameters, it would definitely be interesting to: explore the PageRank way of augmenting the seed words more deeply; generate other poems with a different structure than sonnets, possibly with a predefined rhyme pattern; and to explore the Contextualizer to provide some insights on the contents of the poem, useful to frame it and possibly to evaluate it.

The reported effort constitutes evidence that PoeTryMe can indeed be extended to operate in languages other than Portuguese. The evidence provided by a Spanish instantiation is limited, given the close similarity between the two languages. However, the adaptations required were in no way made easier by those similarities. The possibility of extending the platforms is only restricted by the availability of the lexical, semantic and grammatical resources described, by the existence of a certain affinity between the definition of poetry in the target language (such as being based on length in syllables and rhyme), and by the availability of software solutions for scansion of the desired metrics.

Acknowledgements

This work was supported by projects PROSECCO and ConCreTe. Part of this work was developed during a short term visit funded by the PROSECCO CSA project, Euro-

pean Commission under FP7 FET grant number 600653. The project ConCreTe acknowledges the financial support of the Future and Emerging Technologies (FET) programme within the Seventh Framework Programme for Research of the European Commission, under FET grant number 611733.

References

- Agirrezabal, M.; Arrieta, B.; Astigarraga, A.; and Hulden, M. 2013. Pos-tag based poetry generation with wordnet. In *Proceedings of the 14th European Workshop on Natural Language Generation*, 162–166. Sofia, Bulgaria: ACL Press.
- Barbieri, G.; Pachet, F.; Roy, P.; and Esposti, M. D. 2012. Markov constraints for generating lyrics with style. In *Proceedings of 20th European Conference on Artificial Intelligence (ECAI)*, Frontiers in Artificial Intelligence and Applications, 242, 115–120. IOS Press.
- Boden, M. 1990. *Creative Mind: Myths and Mechanisms*. London: Weidenfeld & Nicholson.
- Brin, S., and Page, L. 1998. The anatomy of a large-scale hypertextual web search engine. *Computer Networks* 30(1-7):107–117.
- Charnley, J.; Pease, A.; and Colton, S. 2012. On the notion of framing in computational creativity. In *Proceedings of the 3rd International Conference on Computational Creativity*, ICCO 2012, 77–81.
- Colton, S.; Goodwin, J.; and Veale, T. 2012. Full FACE poetry generation. In *Proceedings of 3rd International Conference on Computational Creativity*, ICCO 2012, 95–102.
- Colton, S.; Pease, A.; and Ritchie, G. 2001. The effect of input knowledge on creativity. In *Proceedings of the IC-CBR'01 Workshop on Creative Systems*.
- Gervás, P. 2000a. A logic programming application for the analysis of Spanish verse. In *1st International Conference on Computational Logic*, 1330–1344.
- Gervás, P. 2000b. WASP: Evaluation of different strategies for the automatic generation of spanish verse. In *Proceedings of AISB'00 Symposium on Creative & Cultural Aspects and Applications of AI & Cognitive Science*, 93–100.
- Gervás, P. 2001. An expert system for the composition of formal spanish poetry. *Journal of Knowledge-Based Systems* 14:200–1.
- Gervás, P. 2013. Computational modelling of poetry generation. In *Proceedings of the AISB'13 Symposium on AI and Poetry*, 11–16.
- Gonçalo Oliveira, H.; Antón Pérez, L.; Costa, H.; and Gomes, P. 2011. Uma rede léxico-semântica de grandes dimensões para o português, extraída a partir de dicionários electrónicos. *Linguística* 3(2):23–38.
- Gonçalo Oliveira, H.; Cardoso, F. A.; and Pereira, F. C. 2007. Exploring different strategies for the automatic generation of song lyrics with Tra-la-Lyrics. In *Proceedings of 13th Portuguese Conference on Artificial Intelligence*, EPIA 2007, 57–68. Guimarães, Portugal: APPIA.
- Gonçalo Oliveira, H. 2012. PoeTryMe: a versatile platform for poetry generation. In *Proceedings of the ECAI 2012 Workshop on Computational Creativity, Concept Invention, and General Intelligence*, C3GI 2012.
- Gonzalez-Agirre, A.; Laparra, E.; and Rigau, G. 2012. Multilingual central repository version 3.0. In *Proceedings of the 8th International Conference on Language Resources and Evaluation*, 2525–2529. ELRA.
- Manurung, R.; Ritchie, G.; and Thompson, H. 2012. Using genetic algorithms to create meaningful poetic text. *Journal of Experimental & Theoretical Artificial Intelligence* 24(1):43–64.
- Manurung, H. 1999. A chart generator for rhythm patterned text. In *Proceedings of 1st International Workshop on Literature in Cognition and Computer*.
- Manurung, H. 2003. *An evolutionary algorithm approach to poetry generation*. Ph.D. Dissertation, University of Edinburgh.
- Netzer, Y.; Gabay, D.; Goldberg, Y.; and Elhadad, M. 2009. Gaiku: generating haiku with word associations norms. In *Proceedings of the Workshop on Computational Approaches to Linguistic Creativity*, CALC'09, 32–39. ACL Press.
- Oulipo, A. 1981. *Atlas de littérature potentielle*. Number vol. 1 in Collection Idées. Gallimard.
- Padró, L., and Stanilovsky, E. 2012. Freeling 3.0: Towards wider multilinguality. In *Proceedings of the Language Resources and Evaluation Conference*, LREC'12. Istanbul, Turkey: ELRA.
- Queneau, R. 1961. *100.000.000.000.000 de poèmes*. Gallimard Series. Schoenhof's Foreign Books, Incorporated.
- Toivanen, J. M.; Järvisalo, M.; and Toivonen, H. 2013. Harnessing constraint programming for poetry composition. In *Proceedings of the 4th International Conference on Computational Creativity*, ICCO 2013, 160–167. The University of Sydney.
- Veale, T. 2013. Less rhyme, more reason: Knowledge-based poetry generation with feeling, insight and wit. In *Proceedings of the International Conference on Computational Creativity 2013*, 152–159.
- Wong, M. T., and Chun, A. H. W. 2008. Automatic haiku generation using VSM. In *Proceeding of 7th WSEAS International Conference on Applied Computer & Applied Computational Science*, ACACOS '08.
- Yan, R.; Jiang, H.; Lapata, M.; Lin, S.-D.; Lv, X.; and Li, X. 2013. I, poet: Automatic chinese poetry composition through a generative summarization framework under constrained optimization. In *Proceedings of 23rd International Joint Conference on Artificial Intelligence*, IJCAI'13, 2197–2203. AAAI Press.