Natural Language Engineering **1** (1): 000–000. Printed in the United Kingdom

Multilingual Extension and Evaluation of a Poetry Generator

Hugo Gonçalo Oliveira[1], Raquel Hervás[2], Alberto Díaz[2], Pablo Gervás[2,3]

[1] *CISUC, Dep. Engenharia Informática, Universidade de Coimbra, Portugal*
[2] *Dep. de Ing. de Software e Int. Artificial, Universidad Complutense de Madrid, Spain*
[3] *Inst. de Tecnología del Conocimiento, Universidad Complutense de Madrid, Spain*
hroliv@dei.uc.pt, {raquelhb,albertodiaz}@fdi.ucm.es, pgervas@sip.ucm.es

(*Received 19 June 2017*)

Abstract

Poetry generation is a specific kind of natural language generation where several sources of knowledge are typically exploited to handle features on different levels, such as syntax, semantics, form or aesthetics. But although this task has been addressed by several researchers, and targeted different languages, all known systems have focused on a limited purpose and a single language. This article describes the effort of adapting the same architecture to generate poetry in three different languages – Portuguese, Spanish and English. An existing architecture is first described and complemented with the adaptations required for each language, including the linguistic resources used for handling morphology, syntax, semantics and metric scansion. An automatic evaluation was designed in such a way that it would be applicable to the target languages. It covered three relevant aspects of the generated poems: the presence of poetic features, the variation of the linguistic structure, and the semantic connection to a given topic. The automatic measures applied for the second and third aspect can be seen as novel in the evaluation of poetry. Overall, poems were successfully generated in the three languages addressed. Despite minor differences in different languages or seed words, poems revealed to have a regular metre, frequent rhymes, to exhibit an interesting degree of variation, and to be semantically associated with the initially given seeds.
Keywords: poetry, computer-generated poetry, poetry generation architecture, multilinguality, computational creativity, natural language generation

1 Introduction

The development of poetry generation systems has seen a significant increase over the past ten years. This involves not just systems that rewrite simple texts using a poetic form, but a good number of knowledge-based systems, developed as part of research efforts, reported in papers at scientific events, with details of the procedures employed explained, and compared with prior work, according to academic principles. Given the traditional association between poetry and creativity, poetry generation is a very popular task among the Computational Creativity community (Colton and Wiggins, 2012). Usually seen as a particular kind of knowledgeintensive natural language generation, the resulting text exhibits poetic features, such as a regular metre, rhymes, the use of alliteration or figurative language, and has typically a creative value, in addition to other aspects of natural language. For this purpose, poetry generation systems exploit several sources of knowledge: different kinds of corpora, lexicons, knowledge bases and rules that operate at different levels of language.

Despite a clear interest in establishing appropriate reporting procedures for the automatic generation of poetry, there has been very little work on whether the various techniques that have been considered perform equally well when applied to generate poetry across different languages. Different languages are governed by different poetic traditions and thereby subject to different constraints on form, which makes comparisons across languages difficult. Where languages themselves differ in terms of grammar or morphology, comparisons may also become uninformative.

The present paper addresses the previous difficulties at two different levels. First, it explores the possibilities of applying a similar architecture for poetry generation across three different languages. With few exceptions, previous attempts to poetry generation were not concerned with reuse and were thus not conceived as modular frameworks. Second, this paper presents an attempt to establish a set of procedures for the evaluation of automatically generated poetry that can be applied across different languages. The proposed evaluation measures are based on standard linguistic processing techniques that are reasonably language independent – such as n-gram processing and distributional similarity – in the hope that they will allow comparison across languages with little loss of information. Although this forces the evaluation to focus on a subset of the possible aspects that might be evaluated for computer generated poetry, it provides the means for well-founded empirical judgements on the differences that may arise from a shift in language.

The generalization of poetry generation is made possible by reusing a flexible modular architecture, PoeTryMe, previously presented as such (Gonçalo Oliveira, 2012), but then only used for Portuguese. Despite the later adaptation of PoeTryMe to Spanish (Gonçalo Oliveira, Hervás, Díaz, and Gervás, 2014), the current paper goes further and, not only confirms that PoeTryMe can be adapted to non-romance languages, in this case, English, but it also dissects the effort involved in each adaptation, with a focus on exploited linguistic resources and their properties.

The presented evaluation covers three basic aspects that should be measured in poetry generation systems, namely: conformance with a given poetic form, degree of variation across the output itself, and suitability of the output for a given purpose – stated as a set of keywords (hereafter, seeds) to which the poem should relate. Each of the three target aspects is assessed automatically, with well-known measures. To our knowledge, two of them are novel, in the sense that they have never been used for evaluating automatically generated poetry.

The remainder of this paper starts by reviewing previous work on the automatic generation of poetry (section 2). PoeTryMe, the poetry generation platform used throughout the paper, is then described with a focus on its modular architecture (section 3). Section 4 reports the effort of adapting PoeTryMe to three different languages – Portuguese, Spanish and English. In addition to minor adaptations to the existing modules, this involved the integration of available language resources for each of the languages. Section 5 focuses on evaluation. Examples of generated poems are first presented and briefly described. After that, a set of generated poems is evaluated from the point of view of the poetic form, output variation and goal compliance. Section 6 concludes with a final discussion.

2 Related Work

Automated poetry generation has been a growing field for some time. It is a particular kind of natural language generation (Reiter and Dale, 2000) where, in addition to other aspects of natural language, the resulting text exhibits poetic features, such as a regular metre, rhymes, the use of alliteration or figurative language, and has typically a creative value, similarly to humor generation (Valitutti, Doucet, Toivanen, and Toivonen, 2016) and other approaches to linguistic creativity (Veale, 2012).

Previous work is reviewed in this section according to the following three axes: set of languages for which poems have been generated, computational approaches in poetry generation systems, and evaluation of poetry generation systems.

2.1 Set of Languages for which Poems have been Generated

Efforts to generate poetry automatically have been carried out in a number of languages. The majority of the systems developed addressed English and ranged from the generation of simple haikus (e.g. Wong and Chun (2008); Netzer, Gabay, Goldberg, and Elhadad (2009)), to more complex user-given forms (e.g. Manurung (2003); Colton, Goodwin, and Veale (2012)), including also song lyrics (e.g. Barbieri, Pachet, Roy, and Esposti (2012)). There is also a minority of known approaches for generating poetry in other languages, some of which very different, not only in terms of covered poetry forms, but also in terms of phonetics, syntax and other linguistic rules.

Spanish was one of the first languages where the automatic generation of poetry was explored in the context of Artificial Intelligence, and related issues were discussed (Gervás, 2000; Gervás, 2001). For Portuguese, another romance language, song lyrics have been automatically generated for a given melody (Gonçalo Oliveira, Cardoso, and Pereira, 2007), and poetry has been produced according to user-given structures that would set the number of lines, stanzas, syllables per stanza, or the rhyme pattern (Gonçalo Oliveira, 2012),

Traditional eight-line Basque poems, that aim to be sung, have also been produced automatically (Agirrezabal, Arrieta, Astigarraga, and Hulden, 2013). Although, as Portuguese and Spanish, Basque is spoken in the Iberian Peninsula, it has different origins and is significantly different from romance languages. Toivanen, Toivonen, Valitutti, and Gross (2012) have produced poetry in Finnish with the particularity of exploiting two corpora: one for providing semantic content and another for collecting poetic forms.

Asian languages have also been targeted, some of which with specific tonal and rhythm requirements in poetry generation. This includes the generation of song lyrics in Tamil (Ramakrishnan A, Kuppan, and Devi, 2009), a phonetic language; ancient Chinese classic poetry (Yan, Jiang, Lapata, Lin, Lv, and Li, 2013; Zhang and Lapata, 2014), with strict tonal and rhythm requirements; follow-up lines in Bengali (Das and Gambäck, 2014), matching the rhythm of a user-given line; or poetry inspired by news articles, in Indonesian (Rashel and Manurung, 2014).

The previous systems seem to have been specifically-tailored for a single language. As far as we know, there have been no language-independent attempts to poetry generation nor systems adapted to different languages. The closest is probably the architecture of Colton et al. (2012) and the constraint satisfaction approach of Toivanen, Järvisalo, and Toivonen (2013), both originally applied to English, but roughly combined in the system of Rashel and Manurung (2014), which targets Indonesian.

2.2 Computational approaches in poetry generation systems

There have been numerous efforts to model the generation of poetry using Artificial Intelligence techniques. Most of these efforts can be seen as solutions based on the reuse of existing text, subject to certain modifications or reorganizations. The main differences between them lie on the degree of modification of the source text and the choice of unit of recombination during the process of adaptation.

Solutions based on direct reuse of text fragments from existing sources, recombined into different orders, range from the combinatorial approach of Queneau (1961) – which managed to obtain a large number of new poems in French by interchanging the lines of a set of poems, always respecting the same relative position of each line within a set template stanza – to the reuse of complete tweets, or sentences extracted from them, as featured in the FloWr system (Charnley, Colton, and Llano, 2014) which generated English verses. Yan et al. (2013) also reuse text from a large poem corpus, but they formulate poetry generation as an optimization problem based on generative summarization. Their approach involves the retrieval of candidate lines for a user query, their segmentation into constituent terms, term clustering, and selection of sentences that conform the structural constraints, using terms from different clusters. A similar approach is followed by Carolyn E. Lamb and Clarke (2015) and Malmi, Takala, Toivonen, Raiko, and Gionis (2016). Because these solutions reuse text directly, they are applicable to any language. However, they are also restricted in that they take almost no advantage of basic principles of language such as compositionality: they simply search for combinations of existing fragments of text, with no ability to create new interesting instances of similar fragments. We consider this to be a poor solution.

Slightly more elaborate solutions reuse material from known sources but replace specific words with material from a different source. The first known example are the French *rimbaudelaires*, by Oulipo (1981), which used words mined from Baudelaire's poems to replace particular positions in verses from Rimbaud, subject to metrical match between the elements in each substitution. Toivanen et al. (2012) combined two different corpora, one – fragments selected from a corpus of old Finnish poetry

– to mine for verse drafts that were converted into templates by leaving out certain keywords, the other – a word association network built from the Finnish Wikipedia – to obtain candidate words to plug the resulting gaps. In later work (Toivanen, Gross, and Toivonen, 2014), the replacement process is further constrained with restrictions on words already associated with prior knowledge on the poem's topic. This approach of filling templates combined with various restrictions is elaborated further by Colton et al. (2012). To drive this process, they additionally introduce emotional constraints, in the form of a mood – from the analysis of newspaper articles for a particular day – that influences decisions taken in the construction of poems in English. Solutions based on substitution of particular words in existing fragments of text start to introduce linguistic elements into the construction process, and take advantage of the resulting flexibility. Because they rely on basic linguistic information – lexical category, morphological information and metric scansion of both the words being replaced and the words being used in their place – they require language specific functionality in the form of part-of-speech (POS) tagging, solutions for morphological inflection and solutions for metric scansion. In this sense, they are slightly more difficult to generalise to other languages than the solutions based on direct reuse of text fragments. However, POS tagging and morphological flexion are relatively easy to obtain for most languages. Metric scansion presents more difficulty, as different languages rely on different metric traditions, and some of these are more difficult to automate than others.

The process of stripping down a reference text to leave gaps that can be refilled with material from a different source can be taken to the extreme when all the words from a reference fragment are eliminated, retaining only the corresponding sequence of POS tags, to be filled in with words from a different source, selected based on POS tag match and possibly further aptness criteria. This is the approach followed by systems such as the early version of the WASP system (Gervás, 2000) – which combined the structure of known poems with vocabulary provided by the user to generate Spanish poems –, the work of Agirrezabal et al. (2013) – which used both syntactic and semantic information to ensure that structures from Basque poems of different line lengths were filled in with particular attention to the correct transfer of morphological information, extremely relevant in the case of the Basque language. A similar approach is applied to Finnish poetry (Toivanen et al., 2013), where a complex set of constraints is also imposed. As a more elaborate version of the approach discussed previously, the replacement of every word in a given fragment by words deemed appropriate requires much the same set of language-specific functionalities. When metric scansion has to be carried out over fragments of text larger than a single word, the differences across languages may become more marked, due to different traditions with respect to metric phenomena such as synaloepha. Depending on the tag set chosen, simple POS tag preservation during substitution over larger fragments of text may not guarantee grammaticality. It is also likely to result in texts of poor idiomatic correctness.

The technique of Case-Based Reasoning (CBR) (Aamodt and Plaza, 1994) was used in an evolution of the WASP system (Gervás, 2001), which matched a user input with a prose rendition of a poem, and reused the structure of the poem –

in terms of its POS tags – together with the words from the user input. Such an approach basically relies on a case base that encodes transformation between two different styles of writing – prose and poetry – for a given language. The construction of such case bases for a new language requires a substantial effort in annotation and alignment. As a result, this approach is inadequate for a solution to be applied across languages.

Besides word-based specification of content combined with emotional aspects, Misztal and Indurkhya (2014) introduce a different way of constraining form and combining the modules that deal with different aspects of poetry generation. It relies on a multi-agent approach where each module is a set of artificial experts, focused on a particular aspect, that interact by sharing results on a blackboard. Types of experts include *word-generating* – that contribute with words matching a given topic or emotion –, *poem-making* – that arrange words in the common pool into phrases or sentences, guided by Context-Free Grammars –, and *evaluating* experts. Content is constrained in terms of a particular topic and an emotion extracted from the input text, while form is constrained through the implementation of the various experts. In this approach, specific literary forms are introduced explicitly by the set of system modules, rather than arising from the reuse of an existing corpus of poetic texts. This type of system allows for consideration of a very rich set of features – such as emotion or grammar. However, the addition of each of these features requires the implementation of the particular expert module. In most cases, these modules will have to be language-specific, and possibly require a number of resources to inform/encode the behaviour of the feature for that particular language. For these reasons, we do not consider this approach to be particularly appropriate for application across languages.

At the other extreme of the spectrum lie systems which start from an actual message to be conveyed by the poem, specified in terms of a semantic representation. Manurung (1999) pioneered this kind of system by applying a generate & test approach based on chart generation, which received as input a specification of the target semantics in first-order predicate logic (FOL) and a specification of the desired poetic form in terms of metre. Systems of this type require resources to encode or inform the transitions between a FOL representation and a set of grammatical sentences. A chart was built incrementally using words from a lexicon that subsumed the input semantics, and partial solutions were semantically checked, to avoid incompatibility with the original input and ensure compatibility with the desired poetic form. A similar approach, but based on mirroring the meaning of a given textual document, was employed more recently (Tobing and Manurung, 2015). In addition to the poetic features, the system tries to keep the predicate-argument structure of the document. But the authors of these efforts conclude that dealing with so many constraints is computationally impractical. The automated transcription from FOL representation to grammatical sentences is an unsolved open problem in the field of natural language generation, even for those languages for which it has been addressed empirically. Solutions of this type capable of operating across a number of languages would require a very substantial effort of FOL to sentences encoding for each language involved.

A final approach to reuse of existing texts foregoes syntactic structure as featured by POS tag sequences and relies instead on the use of n-grams to model the probability of certain words following others. This can be understood as a process of reusing corpus fragments of size n, to be combined into larger fragments based on the probability of the resulting sequence. Such a procedure is used by the Poetic Machine (Das and Gambäck, 2014) to build single poetic lines matching a given rhyme pattern to be matched, by a redesigned version of the WASP poetry generator (Gervás, 2013a,b) that used an evolutionary programming approach to model the poet's ability to iterate over a draft, and by the work of Barbieri et al. (2012), which relied on Constrained Markov Processes to generate texts as lyrics in the style of an existing author, integrating constraints on grammaticality, rhyme, meter, and, to a certain extent, semantics, into the search procedure itself. N-gram based language models have shown great applicability in many contexts – such as machine translation, speech processing, text prediction or dialog systems – and they present the important advantage of being language independent to a great extent. At least in the sense that they can be trained on large volumes of text in whatever language is required. These solutions do apply the principles of articulation of language, which gives them significant expressive power. However, they take no advantage of other basic principles such as lexical categories or grammar.

All the different approaches to the computational generation of poetic text illustrate a variety of possible ways of partitioning and recombining existing sources of poetic and non-poetic texts so they can be recombined into new poems.

2.3 Evaluation of Poetry Generation Systems

The field of computational creativity is progressively evolving beyond the early stages where acquisition of valid samples of a given style of artefact was acceptable without further consideration, to a point where evaluation of the outcome has become an important requirement (Jordanous, 2012). This is seen as a sign of scientific maturity of the field.

Due to the underlying difficulty of evaluating creative artefacts, and claiming that the intended audience of poetry consists of people, the evaluation of computer generated poetry has often resorted to human judges, who assess produced poems according to a set of predefined dimensions.

In his thesis, Manurung (2003) defined three fundamental properties that should be satisfied by poetic text, namely: meaningfulness – a poem must convey a conceptual message, meaningful under some interpretation –, grammaticality – a poem must obey linguistic conventions prescribed by a given grammar and lexicon – and poeticness – a poem must exhibit poetic features, such as a regular metre or the presence of rhymes. While those properties can sometimes be validated by the methods applied (Manurung, 2003; Misztal and Indurkhya, 2014), they can also be assessed by the observation of the obtained results. In fact, some researchers (Yan et al., 2013; Das and Gambäck, 2014; Zhang and Lapata, 2014) evaluated the output of their system based on the opinion of human judges on set of poems, who answered questionnaires designed to capture Manurung's properties. Still relying on human opinions, other authors got conclusions on the quality of their results with questions that rated slightly different dimensions, though with some overlap. Those include the typicality as a poem, understandability, quality of language, mental images, emotions, and liking (Toivanen et al., 2012); or structure, diction, grammar, unity, message and expressiveness (Rashel and Manurung, 2014).

Among the systems with outputs evaluated by humans, some ended up conducting a Turing test-like evaluation, where the scores of the systems produced by their poems were compared to those for human-created poems (Netzer et al., 2009; Toivanen et al., 2012; Agirrezabal et al., 2013; Rashel and Manurung, 2014). Despite also relying on human evaluation, other researchers compared poems produced only by their systems but using different parameters or strategies (Gervás, 2000; Gonçalo Oliveira et al., 2007; Barbieri et al., 2012; Yan et al., 2013); or poems produced by other systems with a very similar purpose (Zhang and Lapata, 2014).

Established methods to evaluate human creativity, from the psychology domain, have also been proposed to assess computational creativity, including automatically generated poetry. van der Velde, Wolf, Schmettow, and Nazareth (2015) present a map of words related to creativity, obtained from an association study. Among others, these words may be used to define the dimensions to evaluate the creativity of a poem and its creation process. Another example is Lamb, Brown, and Clarke (2016), who relied on a group of human experts to rate the creativity of 30 poems. Some poems were written by human authors and others generated automatically, by different creative systems, though judges were not informed of this. Despite some consensus on the best and worst poems, judges disagreed on the remaining, which made the authors unsure on the suitability of their approach for computer-generated poetry.

There has been a huge discussion on the suitability of the Turing test for evaluating computational creativity approaches (Pease and Colton, 2011). The main criticism is that it is focused on the resulting products and not on the involved creative process, which encourages the application of simpler processes, some of which might be merely concerned with tricking the human judge into thinking their outputs were produced by a human.

In addition to the previous issues, human evaluation of poetry in different languages would pose an additional difficulty. It would either require a group of judges that were fluent enough in each target language, or different but comparable groups for each languages, both highly unlikely to achieve. Though somehow limited, a possible solution is to design an automatic evaluation focused on a subset of equivalent aspects that can be measured in poems written in any of the target languages.

Besides the evaluation of the form constraints (metre and rhymes), few approaches have tried to evaluate poetry generation systems or their results automatically. The previous evaluation is often part of the generation process, as it happens with Misztal and Indurkhya (2014)'s or Gervás (2013a)'s automatic experts, or with many other systems that assess the metre, rhymes and other properties of produced texts during generation time.

Notable exceptions on the automatic evaluation of poetry generation systems include the applications of metrics typically used in the scope of automatic summarization and machine translation, such as ROUGE, to access the performance of Yan et al. (2013)'s system, which is based on a generative summarization method; or BLEU, to assess the ability of Zhang and Lapata (2014)'s system to generate valid sequences of lines.

3 PoeTryMe: a Poetry Generation Platform

PoeTryMe (Gonçalo Oliveira, 2012; Gonçalo Oliveira and Cardoso, 2015) is a poetry generation platform, on the top of which different approaches for poetry generation can be implemented. It was originally designed to test different settings in the process of poetry generation, with a focus on the exploited knowledge resources. This resulted in a modular architecture (see Figure 1) that 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 or developer. Among other parameters, the user may define the rules of the generation grammar, the semantic network to use, the structure of the poem, the set of seed words, the polarity lexicon and the transmitted sentiment. As for the developers, they may re-implement some of the modules and reuse the others.

The modular architecture enables to easily test different settings of parameters and to study their impact in the resulting poems, with reduced effort. Each different setting consists of a new instantiation of PoeTryMe, typically driven towards a different goal than previous instantiations. The work involved for a new instantiation is variable. It might be a matter of changing the underlying resources or initial parameters, such as the target poetry form; it might involve plugging in an additional layer for selecting the seeds (Gonçalo Oliveira, 2016) or generating the semantic network according to some stimuli (Gonçalo Oliveira and Oliveira Alves, 2016); or it might include the implementation of a different Generation Strategy. In the present work, our goal is to generate poetry in different languages, and most of the involved effort, further described in section 4, is on finding, adapting and plugging in adequate underlying resources.

PoeTryMe's architecture has two core modules – the Line Generator and the Generation Strategy – and several complementary modules. To better understand the main goal of this work, in Figure 1, the language-dependent components of the architecture are surrounded by dashed lines. It should be stressed that those components are static knowledge resources, quite standard and available in several languages (semantic network, polarity lexicon, morphology lexicon), or that might be extracted automatically from available text corpora (grammars). These resources are processed by language-independent modules. The only exception is the Syllable Utils, which might require slightly different algorithms for different languages, though all sharing a similar interface. All modules of this architecture are described along this section.

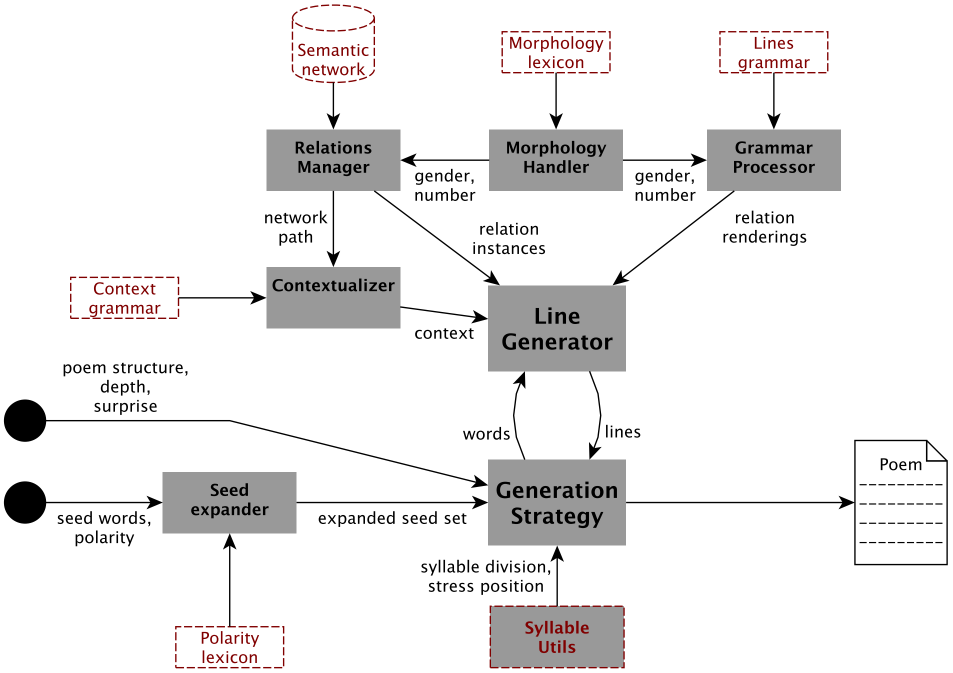

Fig. 1. The architecture of PoeTryMe.

3.1 Line Generator module

The Line Generator module produces semantically-coherent natural language fragments, with the help of:

- A semantic network, managed by the Relations Manager, that connects words according to relation predicates (see Figure 2 for a very simple network, centered on the words *adapt* and *language*).
- A line grammar, handled by the Grammar Processor, with possible ways of expressing semantic relations in text (hereafter, renderings), to be used in the generation of lines (see Figure 3 for a very simple example of a valid rule set, with eight renderings for different semantic relations).

To produce a single line, the Line Generator follows four steps:

1. Select a random relation instance from the semantic network (e.g. *adapt* verb-synonym-of *conform*);
2. Retrieve a rendering for this relation from the grammar, which is possible because there are renderings for each covered relation type (e.g. `<arg1> , my children , and you shall <arg2>`);
3. Insert the relation arguments in specifically marked placeholders of the rendering (e.g. *adapt, my children, and you shall conform*);
4. Return the resulting fragment.

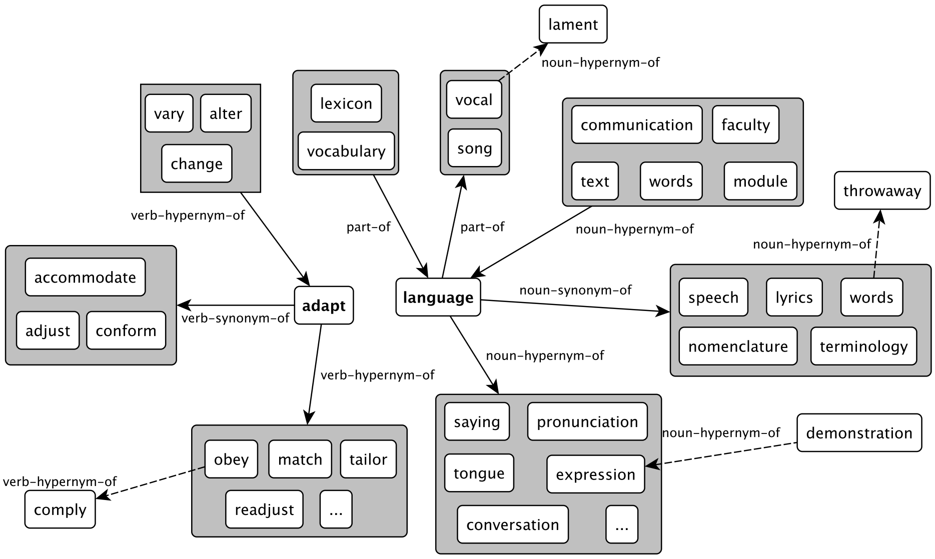

Fig. 2. Semantic Network example

```
NOUN-HYPERNYM-OF   → <arg1> in softest <arg2> flew by
NOUN-HYPERNYM-OF   → by no <arg2> <arg1>
PART-OF            → somewhere over the <arg2> <arg1> are blue
PART-OF            → dark <arg2> on a dangerous <arg1>
VERB-SYNONYM-OF    → <arg1> , my children , and you shall <arg2>
VERB-SYNONYM-OF    → you <arg1> to <arg2> a clue
VERB-HYPERNYM-OF   → you <arg2> , you <arg1>
VERB-HYPERNYM-OF   → <arg1> no questions , <arg2> no side
```

Fig. 3. Grammar example rule set

Generation can be constrained by a set of words (hereafter, seeds), which will result in using only a part of the semantic network in this process, defined by the seeds and their surroundings. The modules involved in this process are described next.

3.1.1 Relations Manager module

The Relations Manager is an interface to the semantic network. It may be used to retrieve all words related to another, or to check if two words are related while providing their relation. Relation instances are represented as triplets like $triplet = \{arg1, RELATION, arg2\}$ that connect two arguments (nodes of the network) with a relation type (labeled edges).

To narrow the space of possible generations, a set of seed words may be provided to the Relations Manager, in order to define the generation domain. This is represented by a subgraph of the main network, where the relation triplets should either contain one of the seed words or somehow related words. More specifically, the subgraph will only contain triplets with words that are at most δ nodes far from a seed word, where δ is a 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. Both δ and ν are given as input parameters. Figure 2 shows mostly words that are one level far from *adapt* and *language*. For illustrative purposes, four words that are two levels far are also present, namely: *comply*, *lament*, *throwaway* and *demonstration*.

3.1.2 Grammar Processor module

The Grammar Processor is an interface to the generation (context-free) grammar. It reads a grammar and, similarly to Manurung (1999), performs chart generation with a chart-parser in the opposite direction. The grammar rules are written in an editable text file, and their body should consist of natural language renderings of semantic relations. So, 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, the rules body may include nonterminal tokens, which can either be references to other rules or placeholder tokens that indicate the position of the relation arguments (`<arg1>` and `<arg2>`), to be filled by the Line Generator. This way, given a relation predicate, the Grammar Processor can retrieve one (or several) renderings for any triplet of that kind. To some extent, the renderings can be seen as n-grams where two words are replaced, constrained by a semantic relation that must be held between the replacement words.

A very simple example of a valid rule set is shown in Figure 3. In this case, each rule has two placeholder tokens and none has a reference to other rules. This makes these specific rules flat and similar to lines templates. The simple grammar contains renderings for the relations of hypernymy (between nouns and between verbs), part-of and synonymy (between verbs). Those rules could be used to generate fragments such as: "*vocal in softest lament flew by*", "*dark lexicons on a dangerous language*", or "*adapt no questions, obey no side*".

3.1.3 Morphology Handler module

The Morphology Handler is an interface to a morphology lexicon. It can be used to adjust the words according to morphological properties that suit their position in the lines, such as gender (masculine or feminine) or number (singular or plural).

3.1.4 Contextualizer module

The ability to explain how its artefacts are created is an important feature of a creative system, which can help the reader or judge to better understand the choices made. PoeTryMe provides this feature by keeping track of all the relation triplets 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 triplets used to form the lines and how they are connected to a word in the generation domain. The context, which can also be used for debugging purposes, can be a mere list of triplets or, if a contextualisation grammar is provided, it may consist of a natural language piece of text. The previous grammar is of the same kind as the lines grammar.

3.2 Generation Strategy module

A Generation Strategy follows an algorithm for organizing produced lines such that they suit, as much as possible, the structure of a poetic form and exhibit certain poetic features. An input structure file contains the number of stanzas, lines per stanza and syllables in each line of the poem. This file may also use a symbol for denoting the target rhyme for the lines. Figure 4 shows the files for a haiku-like structure, a block of four, and an English sonnet. Both the first and the second are single-stanza structures, but the haiku-like has 3 lines with 5, 7 and 5 syllables, while the block has four lines with 10 syllables each. The sonnet is configured as a poem with 4 stanzas: the first three have 4 lines with 10 syllables, and the last one has 2 lines with 10 syllables. The system considers consonance rhymes and a target pattern of this kind of rhyme can be specified in the structure file. In Figure 4, there is no rhyme pattern specified for the haiku, but each line of the sonnet structure has a symbol that results in the following rhyme pattern: *ABBA CDCD EFEF GG*. When a rhyme pattern is not specified, the Generation Strategy may, for instance, consider that a rhyme occurs when any two lines in the same stanza have the same termination. During generation, the Generation Strategy may also consider alliteration, but this is not explored in this work.

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

```
#block-of-four 10
stanza{line(10);line(10);line(10);line(10)}
```

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

Fig. 4. Structure files for a haiku, a block of four ten-syllable lines, and an English sonnet with a rhyme pattern.

An instantiation of the Generation Strategy module does not generate the lines. It exploits the Line Generator module to retrieve natural language fragments, which might be used as poem lines. Each instantiation implements a strategy that may differ in the number of fragments requested from the Line Generator at any time, and how they are organised into the poem structure, considering for this purpose features like metre, rhyme – based on the Syllable Utils – , coherence between lines or others, depending on the desired goal.

Besides the poem structure file, the Generation Strategy has additional inputs, used to constrain the Line Generator, including a set of seed words. Before generation, this set may be expanded, in order to obtain additional relevant words that are not directly related to the seeds. This is achieved by the Seed Expander module. Expansion can be simple or biased towards an input sentiment (positive, neutral, negative), for which a polarity lexicon is used. In addition to describing the sub-modules used by the Generation Strategy, this section ends with an illustrative example of the strategy used in this work.

3.2.1 Syllable Utils module

As its name suggests, Syllable Utils includes a set of operations on syllables. The underlying logic of this module will depend on the current goal, such as target language. Yet, it must provide an interface at least for, given a word: (i) divide it into syllables; (ii) find the most stressed syllable; (iii) extract the termination, useful to identify rhymes. For instance, given the English word *language*, this module would split it into two syllables – [*lan-guage*] – , inform that the most stressed syllable is [*lan*], and return [-*anguage*] as the termination.

3.2.2 Seed Expander module

The number of seed words is open but, before generation, it can still be expanded with the most relevant words for the given seeds. For this purpose, the Seed Expander includes a personalized version of the PageRank (Brin and Page, 1998) algorithm, which can be run in the same semantic network used by the Line Generator. 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 top-n ranked nodes are selected. For instance, the top-10 ranked words for *language* and *adapt* would be: *accommodate*, *adjust*, *change*, *alter*, *conform*, *match*, *tame*, *focus*, *cultivate*, *transcribe.*

The Seed Expander is also an interface to a polarity lexicon. In order to induce a positive or negative sentiment in the generated poem, this module selects, from the set of expanded words, those with the target polarity. For instance, suppose that the top-10 ranked words for the seed “blue” were: *grim, blueness, gloomy, sexy, color, dark, dejected, low, dye, down.* When generating a poem with the top-4 relevant words with a negative bias, the words *grim*, *gloomy*, *dark* and *dejected* would be added to the seed set. For a positive poem using “blue” as the original seed, the word *sexy* would be added to the seed set, together with the next three positive words in the ranking.

3.2.3 Generate & test example

Despite previous experiments with other strategies (Gonçalo Oliveira and Cardoso, 2015), most instantiations of PoeTryMe (e.g. Gonçalo Oliveira et al. (2014); Gonçalo Oliveira and Oliveira Alves (2016)) used a simple generate & test strategy at the line level. For each line in the target poem structure, this strategy successively generates lines, scores them according to the metre and presence of rhymes, and stops either after a predefined number of generated lines (n), or when a generated line has precisely the target number of syllables and target rhyme, if there is one. The score of a line is 0 if it has the same number of syllables of the target. For each syllable more or less, there is a penalty point. In addition to this, if the line rhymes with one already selected for use, a positive bonus is given.

This strategy is illustrated with a short example of the generation of a block-of-four (second structure in Figure 4). For this purpose, the grammar in Figure 3 was used together with a semantic network that included the fragment in Figure 2, as well as the English seeds *adapt* and *language*. For the sake of clarity, the seed expansion was off. The depth parameters were $\delta = 1$ and $\nu = 0.01$, and the rhyme bonus was set to 2 points. Figure 5 uncovers part of the line generation procedure, followed by the resulting poem and its contextualization. A minority of fragments generated for the first two lines of the block-of-four are shown, together with their number, score and an indication whether they were the best line then generated (Best). Generation would stop after a maximum of n generations or after the best possible score is obtained, which happens in the two examples. Since there is no target rhyme for the first line, the best possible score is 0. For the second line, the best possible score would be to match not only the target number of syllables (10), but also rhyme with the previous line (ending in *-ue*), leading to a score of 2.

The resulting poem had an overall score of 4, because it matched the target number of syllables and included two rhymes. The provided contextualization is a mere list of the relation triplets used for each line.

4 Adapting PoeTryMe to different languages

Our main goal involved using PoeTryMe's flexible architecture to produce poetry in different languages. In other words, the result would be a new instantiation of PoeTryMe for each target language. We recall that the original instantiation of PoeTryMe targeted Portuguese. And given that this work was conducted by Portuguese and Spanish researchers, it seemed natural to target our native languages. The third language was English, due to its universality. As a consequence of the modularity of PoeTryMe's architecture, such an instantiation would not require to re-implement the core modules. Instead, most of the effort was to look for suitable linguistic resources in the target languages and integrating them in the platform. Still, in some cases, interfaces had to be implemented or minor adaptations had to be made, so that the resources could be processed by the existing modules. Those resources, surrounded in dashed lines in Figure 1, include:

- Rules of the **Lines Grammar**, written in the target language, either hand-

Line	Generation	Line	Score	Best
#1	1	*you readjust, you adapt*	-2	Yes
	8	*language lexicons ready to go*	-1	Yes
	13	*fairy expressions of demonstration fall*	-1	No
	84	*somewhere over the language, songs are blue*	0	Yes
	Use:	*somewhere over the language, songs are blue*		
#2	1	*language in softest expressions flew by*	0	Yes
	2	*dark lexicons on a dangerous language*	0	No
	120	*you adapt to accommodate a clue*	2	Yes
	Use:	*you adapt to accommodate a clue*		
#3	**Use:**	*what language and speeches between us lie*		
#4	**Use:**	*language in softest expressions flew by*		

Resulting Poem	Contextualization
somewhere over the language, songs are blue	*language* part-of *song*
you adapt to accommodate a clue	*adapt* verb-synonym-of *accommodate*
what language and speeches between us lie	*language* noun-synonym-of *speech*
language in softest expressions flew by	*language* noun-hypernym-of *expression*

Fig. 5. Poetry generation example with PoeTryMe.

crafted, or discovered from human-created text with the help of a semantic network in the same language. The discovery approach results in rules similar to those in Figure 3 – flat templates, which are fragments where two related words co-occur. The related words become placeholders (`<arg1>` and `<arg2>`) and the fragment is added as a possible rendering for their relation.

- A **Semantic Network**, where words of the target language are connected according to labeled relations. These can be well-known semantic relations, such as synonymy, hypernymy, or meronymy, or relations of any other kind. Of course, it is important that the relations can be expressed in text.
- A **Polarity Lexicon**, where words of the target language are associated with the typical polarity they transmit, generally positive, negative or neutral.
- A **Morphology Lexicon** with properties such as the POS, gender, number and lemma for the words of the target language. This is used for performing word inflection.
- A tool for **syllable division** and **rhyme identification** in the target language, based on a set of rules, a pronunciation lexicon, or both.

The remaining of this section describes the resources exploited for each target language, together with the effort involved in their preparation and integration with PoeTryMe.

4.1 Portuguese Instantiation

A total of 4,518 rules (renderings) were discovered for Portuguese, using the lexical-semantic network CARTÃO 3.5 (Gonçalo Oliveira, Antón Pérez, Costa, and Gomes, 2011), enriched with association relations, from The Leipzig Corpora Collection[1], and similes, extracted from the AC/DC corpora collection (Santos and Bick, 2000). The following textual collections were exploited for this purpose:

- Poems in *Versos de Segunda*, a web portal dedicated to Portuguese poetry[2]. These included mostly classical forms of poetry, especially sonnets and other poems that followed a strict metre, rhythm and rhyme pattern.
- Portuguese song lyrics, transcribed in the scope of project *Natura*[3]. As lyrics tend to follow the rhythm of the song, these poems tend to have a higher degree of freedom, concerning their form, as compared to strict forms of poetry.

CARTÃO is a large lexical-semantic network for Portuguese, extracted automatically from three dictionaries. It covers several relation types, including not just synonymy, hypernymy and meronymy, but also others such as causation, purpose-of, or property-of. Combining CARTÃO with the associations and similes resulted in a Portuguese semantic network with 332,302 triplets.

The Morphology Lexicon LABEL-Lex (Ranchhod, Mota, and Baptista, 1999) was used to inflect Portuguese nouns and adjectives. It contains 938,445 inflected words and their morphological properties.

In order to get the polarity of Portuguese words, we relied on SentiLex-PT02 (Silva, Carvalho, and Sarmento, 2012), a sentiment lexicon compiled from several publicly available Portuguese resources. It contains 82,347 entries (7,014 distinct lemmas), corresponding to words, associated with their morphological properties, including lemma and POS-tag, and predicted sentiment towards human subjects. Of the 82,347 entries, 76,738 were added manually and 5,609 automatically. About 21,000 of them are positive, 54,000 negative and 7,000 neutral.

In order to split words into syllables, to detect the most stressed syllable and to identify the words termination, SilabasPT[4] was used. This tool had been developed originally for the same purpose, in the Tra-la-Lyrics system (Gonçalo Oliveira et al., 2007), and already covered the operations required by the Syllable Utils module.

4.2 Spanish Instantiation

For the Spanish adaptation, mostly described in Gonçalo Oliveira et al. (2014), a total of 1,281 relation textual renderings were discovered after exploiting 395 poems taken from an anthology of Spanish poetry on the web[5]. The previous extraction relied on a semantic network acquired from the Spanish Wordnet, part of the Multilingual Central Repository 3.0 (Gonzalez-Agirre, Laparra, and Rigau, 2012), where Wordnets from the Spanish languages and Princeton WordNet (Fellbaum, 1998)

[1] `http://corpora2.informatik.uni-leipzig.de/`

[2] `http://users.isr.ist.utl.pt/~cfb/VdS/zlista.html`

[3] `http://natura.di.uminho.pt/~jj/musica/lista_transcricoes.html`

[4] `https://code.google.com/p/silabaspt`

[5] `http://www.poemas-del-alma.com/`

are integrated. We obtained 366,126 relational triplets from the relation tables of the Spanish Wordnet, plus 58,052 synonymy pairs from words included in the same synset. After filtering some less relevant relation types (e.g. *rgloss*, *see_also*), we ended up with 102,457 triplets between lemmas.

To perform the inflection of Spanish nouns and adjectives, we used the Spanish dictionary from the FreeLing open suite (Padró and Stanilovsky, 2012), which contains 650,000 inflected word forms and their morphological properties.

In order to get the polarity of Spanish words, we used the ElhPolar Dictionary 1.0 (Urizar and Roncal, 2013), which contains the typical polarity for 5,210 distinct words, 1,899 positive and 3,304 negative. ElhPolar was created from the automatic translation of the positive and negative words in an English polarity lexicon, where ambiguities were manually solved, and was enriched with additional Spanish words highly associated with positive and negative tweets.

Finally, to compute the metric scansion of the poems in Spanish, in terms of syllables, the corresponding module of the WASP poetry generator (Gervás, 2000) was employed. This module is a re-implementation of an original set of rules designed as a logic program (Gervás, 2000). For its integration in PoeTryMe, an interface with the operations required by the Syllable Utils module, and shared by the Portuguese tool, was implemented.

4.3 English Instantiation

For English, a total of 8,303 textual renderings were discovered from about 3,400 poems extracted from the Representative Poetry Online (RPO), a web anthology of poetry by the University of Toronto Libraries[6], using a semantic network acquired from Princeton WordNet 3.0 (Fellbaum, 1998). From the previous source, we used only the poems belonging to the Early Modern English and Present-Day English categories, as much of those in the older categories were written in archaic English.

To extract the semantic network from WordNet 3.0, we followed a similar procedure as for the Spanish network. We first obtained 1.2M relation triplets, which became 696,377 after filtering less relevant relation types (same as those for Spanish) and inverse relations.

In English, there are no gender specific inflections, so, for nouns, only plurals were added. For this purpose, we used the Freeling English noun dictionary, which contains 40,539 entries.

We got the polarity of English words from Bing Liu's Opinion Words (Liu, Hu, and Cheng, 2005), which contains 6,800 pairs of words and their typical polarity, manually compiled over many years. More precisely, it covers 2,012 positive words and 4,788 negative.

On the other hand, metric scansion is more complex for English than for Portuguese and Spanish. While, for the latter, we could cover most cases with a rule-based approach, relying only on the orthography, for English, there are many different combinations of letters that are pronounced the same way (e.g. *eye* rhymes

[6] `http://rpo.library.utoronto.ca/timeline/`

with *lie*, *apply* and *levi*; *air* rhymes with *aware* and *bear*). Therefore, in order to perform syllable division, stress and rhyme identification, we relied on the CMU Pronouncing Dictionary[7], which contains over 134,000 words and their pronunciations in North American English. This involved the development of a specific parser for this dictionary and an implementation of the Syllable Utils interface, to perform the syllable related operations on English words.

This option means that, in contrast to Portuguese and Spanish, we would only retrieve the syllable-related properties of English words that are in the CMU dictionary, which has still a great coverage (more than 133,000 words). For non-covered words, our fallback mechanism uses the Portuguese rules. Still, to minimise this issue, in this work, we removed all non-covered words from the semantic network, and were left with 175,821 triplets. We empirically noticed that this has also a positive effect on the occurrence of rhymes (see Section 5.2).

4.4 Summary

Table 1 summarises the resources and tools used for each language, namely the source of the original poems, the semantic network, the polarity lexicon, the morphology lexicon and the syllables operations tool. For easier reference, we added the number of entries behind the lexicons, and table 2 has qualitative and quantitative information on the semantic networks and generation grammars used for each language. More precisely, it shows the relation types covered, the number of triplets of each type, and the number of renderings in the generation grammar, discovered for each relation type, from the exploited poems, with the help of the semantic network. These numbers show that the resources used for each language are significantly different in terms of size and quality. The Portuguese network is substantially larger than the other two and also covers more relation types. But this does not necessarily result in a larger grammar, as the English grammar contains almost twice the number of renderings of the Portuguese and eight times more than the Spanish. For the three languages, the large majority of the renderings is for synonymy and hypernymy, which are also the relations with more triplets. All of this should make an impact in the poetry generated for each language.

Tools/resources	**Portuguese**	**Spanish**	**English**
Poems	*Versos de Segunda*, Project *Natura*	*Poemas del Alma*	RPO
Semantics	CARTÃO	Spanish wordnet	Princeton WordNet
Morphology	LABEL-Lex *(938k entries)*	FreeLing *(650k entries)*	FreeLing *(40k entries)*
Polarity	SentiLex-PT *(7k entries)*	ElhPolar *(5.2k entries)*	Bing Liu *(6.8k entries)*
Syllables	SilabasPT	WASP	CMU Dict

Table 1. *Summary of tools and resources used for each language.*

[7] `http://svn.code.sf.net/p/cmusphinx/code/trunk/cmudict/`

Relation	**Portuguese**		**Spanish**		**English**	
	Triplets	Renderings	Triplets	Renderings	Triplets	Renderings
Synonymy	135,408	2,292	29,089	346	34,186	1,610
Hypernymy	95,691	928	57,224	536	94,578	4,077
Antonymy	1,538	198	4,625	122	0	0
Part-of	9,637	165	5,110	165	6,288	801
Member-of	8,507	74	0	0	1,015	44
Substance-of	905	15	0	0	896	18
Causation	12,760	64	417	1	580	47
Property-of	38,094	119	5414	86	0	0
State-of	614	5	578	25	0	0
Domain-of	0	0	0	0	14,286	120
Entailment	0	0	0	0	1,228	252
Similar-to	0	0	0	0	22,764	1,334
Manner-of	4,388	104	0	0	0	0
Purpose-of	16,639	120	0	0	0	0
Quality-of	2,407	9	0	0	0	0
Contained-in	683	10	0	0	0	0
Place-of	1,737	13	0	0	0	0
Producer-of	2,436	21	0	0	0	0
Association	858	381	0	0	0	0
Total	332,302	4,518	102,457	1,281	175,821	8,303

Table 2. *Relation types, number of triplets in the semantic networks, and renderings in the generation grammar used for each language.*

5 Evaluation

A creative system should be able to produce outputs that are both novel and meaningful. Its outputs should not be always the same for the same given **input** parameters, but they should, at the same time, have a connection to those parameters. An important obstacle to the evaluation of systems that somehow attempt to mimic human creativity is that criteria for evaluating human creativity are radical and unforgiving: if an aspiring creator produces anything resembling work already done by an earlier creator, he is dismissed as worthless. This makes it extremely difficult to come up with an experimental set up for the human evaluation of computer generated poetry that is not conceptually flawed from the start. If results resemble previous work by human poets, critics evaluating them as human output would score them down. Making evaluators aware of the computer generated nature of the material induces an undesirable bias. If evaluators start allowing for style mimicry, they are no longer applying the same criteria as for human produced material. It is also frequently observed that if critics observe significant discrepancies between computer generated poetry and the existing body of (human produced) poetic works, they tend to consider this an indication of failure on the part of the machine rather than an indication of originality. In view of this, the evaluation described in the present paper focuses on objective measures of the output that might be considered indicative of its quality as a poetic product, regardless of possible human subjectivity.

Under these principles, this section presents an effort to automatically assess the response of the new instantiations of PoeTryMe, one for each target language, to the given input parameters. The multilingual setting posed additional restrictions on the applied procedures, which had to be applicable to each language. It left out aspects such as the intent of the poem, which would probably have to be assessed by humans and face not only the previous issues, but also the difficulties of finding comparable groups of judges for each language. Therefore, we do not necessarily see this evaluation as complete, but we like to see it as a valuable effort to achieve our purpose by focusing on a subset of relevant measurable aspects. Briefly, we aimed at assessing three dimensions, namely:

- Poetic features (Section 5.2): the conformance with the metre or the rhymes, which are distinctive features of poetic texts.
- Structure variation (Section 5.3): the variation across different poems and lines of each poem, to confirm if the system is capable of generating different outputs every time.
- Topicality (Section 5.4): the topic invoked by the poem, which should have a semantic connection to the seed words provided.

Instead of relying on human judges, it was our intention to assess the system automatically, using suitable measures, previously used for other purposes. We hope this to be a contribution for future approaches to the evaluation of particular aspects of poetry generation systems. Although humans are the audience of poetry, we believe that it is not necessary to resort to their subjective opinions every time and dealing with all the related issues, such as finding the adequate number of judges, or the time consumed. In our case, we would have an additional challenge as we would require to either find a comparable group of judges for each language, or a group of judges that would speak all the three, with a similar level of fluency.

The report of each automatic evaluation performed follows the description of the samples used for this purpose in Section 5.1.

5.1 Evaluation Samples

Different languages are governed by different poetic traditions and thereby subject to different constraints on form. And even though some poetic forms are given the same name in different languages, there can be some variations. For instance, the sonnet, one of the most common poetic forms, has fourteen 10-syllable lines but, depending on the language, the traditional rhythm, grouping of lines or rhyme schema is not always exactly the same.

In order to evaluate the different dimensions of the resulting poems, 90 poems were generated for each language. All of them followed a relaxed interpretation of an English sonnet that would suit any language: three blocks of four lines and a final block of two lines, with a free rhyme scheme (see its structure file in Figure 4). Each sonnet was produced by a generate & test strategy, used in previous instantiations of PoeTryMe and briefly illustrated in Section 3.2.3.

The generate & test strategy was used with some fixed parameters, namely:

- Neighbourhood depth (δ) = 1;
- Surprise factor (ν) = 0.0005;

- Maximum generated fragments per target line (n) = 2,000;
- Progressive multiplier = 0.75;
- Rhyme bonus = +2, unless when the word is the same, when it is −1;
- Different length than the target results in a −1 penalty per syllable;
- Alliteration bonus = 0.

The combination of δ and ν means that only words directly related with the seeds were used, or, with a probability of 0.0005, words that were two edges far from them. In the generate & test process, at most 2,000 fragments were produced for the first line of each block, but this number n could increase up to $n + n \times 0.75 \times (i - 1)$, for the ith line in the block. For instance, for the fourth line, $n = 6,500$. Each line is scored according to the absolute difference between its number of syllables and the target number in the poem template. Alliteration was not considered here. The overall score of a poem sums the score of each of its lines plus 2 bonus points for each rhyme.

Two parameters were changed across the experiments: the seeds and the polarity. From an initial set of concepts suggested by each author of this paper, ten were manually selected and translated to words in the three languages. Selection had in mind the inclusion of words of different POS and from different domains. Table 3 shows the selected seeds in the three languages.

#	**Portuguese**	**Spanish**	**English**
1	*amor*	*amor*	*love*
2	*artificial*	*artificial*	*artificial*
3	*azul*	*azul*	*blue*
4	*cantar*	*cantar*	*sing*
5	*computador*	*ordenador*	*computer*
6	*construir*	*construir*	*build*
7	*futebol*	*fútbol*	*football*
8	*ler*	*leer*	*read*
9	*novo*	*nuevo*	*new*
10	*poesia*	*poesía*	*poetry*

Table 3. *Seed words used for generating the poems of the evaluation sample.*

Finally, the seed words were expanded in the following way: for each seed, the four most relevant words according to PageRank were selected, three times considering no polarity, three with positive polarity and three with negative polarity. Of course, since we were using different semantic networks, with different structures, coverages, and created by different means, the additional words for each language were not comparable. For illustrative purposes, Table 4 shows the expansions for the word *artificial* in each language and polarity.

In the end, we had a total of 9 sonnets for each seed word (3 for each selected polarity), and a total of 90 sonnets per language when using the 10 seed words. The poems in Figures 6, 7, 8, 9, 10 and 11 illustrate the evaluation sample. All of them used the seed, the additional relevant words given the polarity (provided in each caption), or words related to the previous. Figures 6 and 7 are examples in Portuguese. The first was generated with the seed 'artificial' and negative polarity and uses negative synonyms of these words, such as *fingida* or *postiça* (in English, fake),

Language	Polarity	**Additional words**
Portuguese	No polarity	*artificialidade, carro, fingida, natural*
	Positive	*natural, sofisticada, concisa, esmerada*
	Negative	*artificialidade, fingida, afectada, teatral*
Spanish	No polarity	*irreal, sintético, falso, natural*
	Positive	*formal, verdadero, afirmación, auténtico*
	Negative	*falso, invención, afectado, ficticio*
English	No polarity	*affected, unreal, unnatural, false*
	Positive	*unreal, colorful, stylized, fabulous*
	Negative	*unnatural, false, contrived, fake*

Table 4. *Additional relevant words for the word artificial, in each language and polarity.*

and also antonyms, such as *natural.* Figures 8 and 9 are examples in Spanish. The first was generated with the seed 'leer' and a positive polarity and uses hyponyms of this verb, including *dictar* (in English, dictate), and other related words, such as *entender* (in English, to understand), which is usually positive, or *adivinar* (in English, to guess). Figures 10 and 11 are examples in English. Instead of our own explanation of the word choice, we put each English poem side-by-side its contextualization, as provided by the Contextualizer module – a list of triplets that explain the semantic connection between each seed word and the other words used. When one of the words used is one level further from a seed, a → is used between the two relevant triplets.

fingida velhaca de eleição
por mais fingida que imitação
foice imune mulher afectada
uma afectada outra buscada

ai aldeia velhaca e fingida
é o meu carro artificial
do artificial ou do natural
nem a afectada do psicasténico

postiça fingida de eleição
por mais fingida que imitação
fingida mais imitação é bom
c'um kartista só de pessoas feito

são psicasténicos de afectada
nos braços da afectada buscada

Fig. 6. Portuguese poem, generated with the seed 'artificial' and negative polarity. Additional relevant words were: fingida (fake), afectada (affected), artificiais (artificials) and artificialidade (artificiality).

ternos e amorosos felizmente
ligeiro afecto vão dependente
na amorosidade amorosa
conviva pessoa boa e gostosa

na amorosidade amorosa
penetra cultas e hidrolatrias
penetra cultas e autolatrias
conviva pessoa boa e gostosa

afecto dedicado sem te ver
cultas e dízimas que como vissem
as liturgias cultas do profundo
c'um passadismo só de cultas feito

as pessoas não têm oportunista
d'outra pessoa noutro avalista

Fig. 7. Portuguese poem, generated with the seed 'amor' and positive polarity. Additional relevant words were: amorosos (lovely), afecto (affection), amorosidade (loveliness) and cultas (cult).

Looking at these samples, and given our knowledge about PoeTryMe, the system outputs can be described according to Manurung (2003)'s definition of poetic text:

- Grammaticality: the lines of the generated poems follow (roughly) grammatical and syntactic rules for each language, which is taken care by the lines grammar and the morphology lexicon;

quiero entender oír sin querer
quiero examinar reconocer
mi pasar lista no quiere leer
quiero adivinar suponer

quiero reconocer examinar
quiero adivinar conjeturar
de leer y dictar la fe de cristo
mi leer no quiere ejecutar

mi adivinar no quiere prever
quiero examinar reconocer
quiero aventurar adivinar
quiero adivinar aventurar

quiero adivinar aventurar
quiero aventurar adivinar

Fig. 8. Spanish poem, generated with the seed 'leer' and positive polarity. Additional relevant words were: examinar (examine), adivinar (guess), reconocer (recognize) and entender (understand).

en afección y achaque partidas
bandos afectas gustos estragados
de cálidos amores y cuidados
bandos afectas gustos estragados

cartas calenturas sobres amores
dulces en el cariño y sufridores
afección afectuosa negro pelo
en su amor cada uno dellos cielo

recios miembros de amores bien dulces
de cariños afectas gallardetes
bandos afectas gustos estragados
en el tierno amor ejercitados

saldré de la emoción y amor
afección de sensaciones y dalias

Fig. 9. Spanish poem, generated with the seed 'amor' and no polarity set. Additional relevant words were: afección (affection), afectas (affected), cariño (endearment) and amores (loves).

into a quicksand what can match the rhyme
and old men tick and young men check their prime
morning tranquilize up still all the boughs
still photographs dark where they cut for bows

to let thy mining hush here unaware
who at the bridge would gibe first to square
persuasion down with a pitiful line
what still unruffled faces what divine

of life's still smooth of death whose mortal threat
nets caught the credit pictures tore the net
with tranquilize and still we saw them go
poetry of time whose stills of deep woe

no setups seize her in their strong mad still
but for the rest still all change all they will

Contextualization:
rhyme HAS-HYPERNYM match
check SAME-GROUP-AS tick,
→ check HAS-HYPONYM rhyme
still HAS-SYNONYM tranquilize
photographs HAS-HYPONYM still
mining DOMAIN-OF hush
gibe HAS-HYPONYM square,
→ rhyme HAS-HYPERNYM gibe
line HAS-HYPERNYM persuasion,
→ line HAS-HYPONYM verse
unruffled HAS-SYNONYM still
smooth HAS-SYNONYM still
pictures HAS-PART credit,
→ still HAS-HYPERNYM pictures
poetry DOMAIN-OF stills
setups HAS-HYPONYM still
still HAS-HYPERNYM change

Fig. 10. English poem, generated with the seed '*poetry*' and no polarity set. Additional relevant words were: *rhyme*, *still*, *hush* and *verse*.

- Poeticness: the lines of the generated poems have a regular metre, frequent rhymes, and are organized in a poetic form, given as input. These features are assessed automatically in Section 5.2;
- Meaningfulness: the generated poems use words semantically-related with the seeds, in semantically-coherent lines. But the latter is not enough to hold the property of meaningfulness. The cohesion provided by using words of the same semantic domain is not always enough for the poem to transmit a clear message as a whole, despite the high level of abstraction. This happens mainly because the generation strategy does not guarantee a logical sequence of lines, and also because different senses of the same words can be mixed. We rather argue that, though poems do not always transmit a clear message, they are related to a certain topic. This aspect is assessed automatically in Section 5.4.

and a night is in the love of the ardor
we can shoot and flick and angle and film
that i might plod and plod and bed the plots
but you who in my voice's fear and worry

to fear the new light closer saint and cling
to crush the human soul subjugate wing
to score him known to man then would man write
let temple fear or flax an equal fright

throughout his crush wild crowds and fearful ran
then drew the shag like the fuck of a man
fuck the long jazz and break the lords of fight
ah score the most of what we yet may write

nor could he fuck to score them off again
crushing charred lines and molten parts of men

Contextualization:
ardor HAS-HYPERNYM love
flick DOMAIN-OF film,
→ film HAS-HYPERNYM object
bed HAS-HYPERNYM plots,
→ fuck HAS-SYNONYM bed
fear HAS-HYPERNYM worry
fear HAS-HYPONYM saint
subjugate HAS-HYPERNYM crush
score HAS-HYPERNYM write,
→ fuck ENTAILS score
fright CAUSES fear
crowds HAS-HYPONYM crush
fuck HAS-SYNONYM shag
fuck HAS-SYNONYM jazz
fuck ENTAILS score
parts HAS-HYPERNYM lines,
→ object HAS-HYPONYM parts

Fig. 11. English poem, generated with the seed '*love*' and negative polarity. Additional relevant words were: *fear*, *crush*, *fuck* and *object*.

5.2 Evaluation of Poetic Features

To evaluate the poetic features of PoeTryMe outputs, we focused on two important features of a poem: the number of syllables per line and the presence of rhymes. For each of the languages – Portuguese (PT), Spanish (ES), English (EN) – Table 5 shows the average ratio of syllables per line and Table 6 shows the average ratio of rhymes per line, both according to the used seed. The first ratio is computed as the sum of the syllables in each line of the produced poems, divided by the total number of lines. Since the number of syllables is not upper-bounded, we present the mean absolute deviation (MAD) around the ideal number, which is always 10 in a sonnet and also happens to be the mode. The rhymes per line is the ratio between the number of lines that rhyme with another in the same poem and the total number of lines in a poem. In this case, the ratio is always between 0 (no rhymes) and 1 (all lines rhyme with another), so averages are presented together with the standard deviation.

Given the line scoring function of the generate & test, lines without the target number of syllables can be used due to two main reasons: either it was not possible to find a suitable line, given the generation parameters, or there was a line with one syllable more or less but that rhymed with a previous line. In our experimentation scenario, the former is unlikely. First, because at least 2,000 lines are generated when a best match is not found. Second, because the majority of the human-created poems used to extract the grammars were sonnets, in all the languages. So there should be more than enough renderings with length close to 10.

On the presence of rhymes, the numbers show that, on average, for any seed, at least half of the lines end up in rhyme. This number is similar for Portuguese and Spanish (0.68 and 0.69), and higher for English (0.76). The higher ratio for English is likely to have benefited from the removal of triplets with words not covered by the CMU dictionary, mostly unfrequent words, some of which with terminations that are harder to rhyme with.

Seed /			$\overline{Syllables/Line}$			
Polarity	**PT**		**ES**		**EN**	
	Means	MAD	Means	MAD	Means	MAD
1	10.05	0.16	10.07	0.13	10.03	0.11
2	10.02	0.07	10.04	0.09	10.06	0.10
3	10.02	0.12	10.00	0.10	9.98	0.13
4	10.04	0.10	9.97	0.08	10.00	0.13
5	9.98	0.10	10.07	0.13	10.02	0.08
6	10.03	0.10	10.00	0.11	10.04	0.09
7	10.02	0.10	10.06	0.09	10.05	0.10
8	10.05	0.16	9.90	0.17	10.03	0.14
9	10.05	0.08	10.07	0.10	10.04	0.09
10	10.02	0.09	10.04	0.09	10.05	0.11
0	10.04	0.11	10.05	0.09	10.04	0.12
+	10.02	0.12	10.03	0.13	10.03	0.10
−	10.02	0.09	9.99	0.11	10.03	0.10
All	10.03	0.11	10.02	0.11	10.03	0.11

Table 5. *Average ratio of syllables per line and mean absolute deviation from the ideal (10 syllables) according to language and seed or polarity.*

Seed /		$\overline{Rhymes/Lines} \pm \sigma$	
Polarity	**PT**	**ES**	**EN**
1	0.70±0.46	0.57±0.49	0.75±0.44
2	0.67±0.47	0.62±0.49	0.67±0.47
3	0.76±0.43	0.76±0.43	0.83±0.38
4	0.71±0.45	0.87±0.33	0.84±0.37
5	0.65±0.48	0.71±0.45	0.71±0.45
6	0.68±0.47	0.79±0.40	0.75±0.44
7	0.68±0.47	0.62±0.49	0.73±0.44
8	0.63±0.48	0.87±0.33	0.86±0.35
9	0.68±0.47	0.52±0.50	0.70±0.46
10	0.67±0.47	0.52±0.50	0.75±0.44
0	0.66±0.47	0.68±0.47	0.78±0.42
+	0.71±0.45	0.68±0.47	0.74±0.44
−	0.68±0.47	0.71±0.45	0.76±0.43
All	0.68±0.46	0.69±0.46	0.76±0.43

Table 6. *Average ratio of rhymes per line and standard deviation, according to language and seed or polarity.*

For Spanish, the three verbs (seeds 4, 6 and 8) were the seeds that lead to a higher rhyme ratio. This happens especially due to the following situations: (i) the termination of verbs in the infinitive is very regular in Spanish, as they always end in *-ar*, *-er* or *-ir*; (ii) PoeTryMe does not inflect the verbs; (iii) using our expansion approach, a verb seed tends to be expanded with other verbs. For instance, with negative polarity, *cantar* (sing) is expanded with *vociferar* (shout), *chirriar* (creak), *traicionar* (betray), *criticar* (criticize), all ending in *-ar*.

For Portuguese, although the verbs have similar constructions, this phenomena is only clear for the seed *cantar* and not for the other verbs. This happens not only due to the higher number of renderings for Portuguese, but especially because there are more cross-categorical relations in the semantic network for this language, which leads to seed expansions with words of different POS. For instance, the expansion of the seed *ler* (read) – *notícia* (news), *leitoras* (readers), *espaço* (space), *deixar* (let) – contains just one additional verb.

Polarity does not have a significant impact on the number of syllables per line. On the other hand, it has some impact on the rhyme ratio but, curiously, the polarity with higher rhyme ratio is different for every language. For Portuguese, it is positive, for Spanish, negative, and for English, no polarity. Although we could go further to analyze these numbers for the seed/polarity combinations, for the sake of clarity, we decided to leave this analysis out of the present work.

To analyze the variation of words in each poem, Table 7 shows the average ratio between distinct words out of all the content words used in a poem[8]. English is clearly the language with more distinct words, followed by Portuguese and Spanish. More than looking at the size of the respective semantic networks, these numbers are explained by the highest number of renderings available for English, about 8,000 against the 4,000 for Portuguese and just about 1,000 for Spanish. For English, the seeds that lead to more distinct words are the four verbs (seeds 4, 6, and 8) and the adjective *blue*. The seeds that lead to less distinct words in this language, *artificial* and *new*, still have a higher average ratio than the highest ratios for Portuguese (0.70 against 0.63), curiously for the seed *novo* (new). In fact, Portuguese seems to have a different behavior than English, as the lower ratios for the former language are obtained with two verbs (*cantar* and *construir*). For Spanish, only the seed *fútbol* leads to an average ratio higher than 0.5. Similarly to Portuguese, the seeds that lead to lower variation are the equivalent verbs (*cantar* and *construir*).

Seed	$\overline{Distinct\_words}$		
	PT	**ES**	**EN**
1	0.55±0.01	0.47±0.06	0.74±0.00
2	0.56±0.05	0.44±0.08	0.70±0.03
3	0.55±0.10	0.46±0.04	0.83±0.02
4	0.47±0.03	0.24±0.03	0.84±0.02
5	0.60±0.06	0.45±0.13	0.76±0.03
6	0.47±0.05	0.32±0.05	0.82±0.05
7	0.57±0.06	0.53±0.03	0.75±0.02
8	0.50±0.07	0.24±0.05	0.79±0.02
9	0.63±0.02	0.43±0.09	0.70±0.05
10	0.54±0.08	0.45±0.03	0.75±0.03
All	0.54±0.08	0.41±0.12	0.75±0.05

Table 7. *Average ratio of distinct words out of all the content words, according to language and seed.*

[8] We considered that content words were all of those remaining after removing stopwords, using the Snowball lists for each language, respectively available from `http://snowball.tartarus.org/algorithms/portuguese/stop.txt`, `http://snowball.tartarus.org/algorithms/spanish/stop.txt`, and `http://snowball.tartarus.org/algorithms/english/stop.txt`

5.3 Evaluation of Structure Variation

To assess the variation of the produced text in terms of structure, we relied on evaluating the structural similarity of the poems in the evaluation samples. For this purpose, we decided to compute metrics that are typically used for evaluating automatic summarization tasks, though in a different way and with a different purpose than Yan et al. (2013). ROUGE (Recall-Oriented Understudy for Gisting Evaluation) (Lin and Och, 2004) is a set of measures to determine the quality of an automatically extracted summary by comparing it to other ideal gold reference texts, created by humans. The covered measures compute the number of overlapping units, such as n-gram, word sequences, and word pairs, between the computer-generated text and the gold references. This metric is appropriate in this case, as we want to compare the structural similarity of poems in terms of n-grams and the appearance of consecutive words in their lines. Therefore, if a pair of poems has a high ROUGE value, it means that the poems have very similar text. If the average ROUGE values for different poems are very high, then the poems generated by the system are too repetitive.

For this evaluation, we have used the following ROUGE measures:

- *ROUGE-N* counts the number of matching n-grams in the compared texts. The n-gram lengths usually range from 1 to 4.
- *ROUGE-L* is based on longest common subsequence (LCS) statistics between a candidate and a reference text.
- *ROUGE-SU-N* considers n-grams that do not have to be consecutive in the text, but could present a maximum of N terms between them, plus unigram-based co-occurrence statistics.

All ROUGE values range from 0 to 1, with 1 meaning that the compared texts are the same and therefore present the same n-grams in the same order. As an example, Table 8 presents the evaluation results obtained for two pairs of compared sentences:

(a)	A big green desk	(b)	A big green desk
	A big green desk		A big red desk

When both sentences are equal (a) all ROUGE measures are 1 as all n-grams are the same. However, when one word is different (b) the change is reflected in the n-gram metrics of ROUGE. For example, with ROUGE-1 all 1-grams are the same except one, with ROUGE-2 only one bi-gram of three is the same, and there are no equal 3- or 4-grams, so ROUGE-{3,4} results are 0.

	ROUGE-					
	1	**2**	**3**	**4**	**L**	**SU4**
a	1	1	1	1	1	1
b	0.75	0.33	0	0	0.75	0.5

Table 8. *Examples for the evaluation metrics*

We first computed ROUGE results for each set of poems from the evaluation sample that had been generated with the same seed and target polarity. Then, inside each of those sets, each poem was compared to each other, and the average value of each metric was computed. When comparing a pair of poems, we used three different configurations:

1. *One-One Comparison*, where each line of the first poem is compared with the line in the same position in the second poem.
2. *One-All Comparison*, where each line of the first poem is compared with all the lines in the second poem.
3. *Whole Comparison*, where two poems are compared as whole texts, not considering the organization in lines.

In addition, a fourth configuration (*One-Alone Comparison*) was considered involving only one poem. More precisely, each poem was compared with itself by comparing each of its lines with all the other lines in the same poem. Figure 12 shows a graphical representation of the four configurations considered.

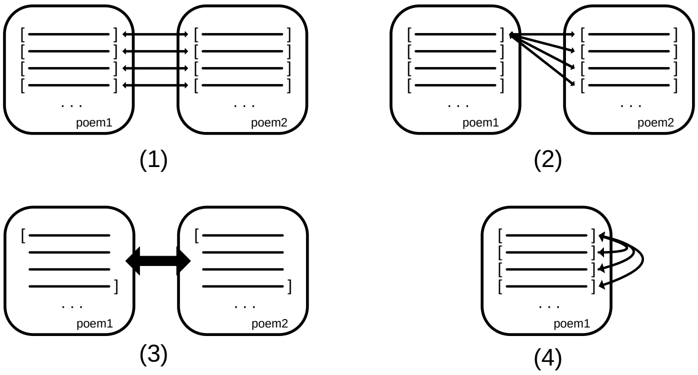

Fig. 12. Configurations used in the evaluation of structure: (1) One-One Comparison; (2) One-All Comparison; (3) Whole Comparison; and (4) One-Alone Comparison

The results are shown in Table 9. Extra experiments were performed for calculating structural similarity of poems with different seeds or polarities, but there were no significant differences in the values obtained, so we have not included these values for the sake of clarity.

The higher values are obtained with the *Whole Comparison* configuration. This means that the poems present a reasonable structural similarity when they are considered as a whole text, not taking into account their lines. In any case, the values obtained for this configuration are not exceedingly high if we consider that a result of 0.50 (only obtained in Spanish) means that half of the n-grams in a poem appear in another one but not necessarily in the same order. The other three configurations, which consider the lines as the units of comparison, present similar results with very low values (less than 0.10 in most cases). We can therefore conclude that the generated poems are not only different among them but varied enough in their words and structure.

	One-One			**One-All**			**Whole**			**One-Alone**		
	PT	**ES**	**EN**	**PT**	**ES**	**EN**	**PT**	**ES**	**EN**	**PT**	**ES**	**EN**
R-1	0.10	0.17	0.08	0.10	0.17	0.08	0.41	0.50	0.31	0.11	0.19	0.08
R-2	0.03	0.05	0.00	0.03	0.06	0.00	0.16	0.25	0.03	0.03	0.07	0.00
R-4	0.01	0.02	0.00	0.01	0.02	0.00	0.06	0.10	0.00	0.01	0.03	0.00
R-L	0.10	0.16	0.08	0.10	0.17	0.07	0.40	0.50	0.30	0.10	0.18	0.07
R-SU4	0.04	0.09	0.02	0.04	0.09	0.02	0.18	0.29	0.09	0.04	0.11	0.02

Table 9. *ROUGE results obtained for the evaluation of the structural similarity of the generated poems*

Differences in the results obtained by the five metrics are expected, with slightly higher values in metrics that use smaller n-grams. For example, the number of individual words appearing in two poems or lines (R-1) will always be higher than the number of 4-grams (R-4 or R-SU4). R-L scores, that are very close to R-1 ones, suggest that, in most cases, the longest common subsequence has a length of one.

From the point of view of the differences across languages, the structural similarities are higher in Spanish, followed by Portuguese and very closely by English. This makes sense due to the lower number of renderings available for Spanish (1,000), which makes poems more "repetitive", whereas there are about 4,000 renderings for Portuguese and 8,000 for English. Also, the smaller size of the Spanish semantic network, 102k triplets, in contrast to the 332k for Portuguese and the 175k for English), makes the vocabulary for Spanish less rich than for the other two languages. Nevertheless, the obtained values are quite low in all cases, meaning that the generated poems are very different from the point of view of their textual structure, a desirable property in a poetry generator.

5.4 Evaluation of Topicality

One of the goals of PoeTryMe is to produce meaningful text with a semantic connection to the given seed words. In other words, the seed should define the topic the poem is about or clearly set its semantic domain. Since the lines of the poems are based on original fragments, where two words are replaced by two other words semantically related in the same way, each fragment should make sense alone. Moreover, the replacing words should be also semantically associated to the seeds, so that a coherent meaning emerges from the poem. To evaluate the semantic similarity between the generated poems and their seeds, we have applied Pointwise Mutual Information (PMI) (Church and Hanks, 1990), a statistical measure typically used to compute distributional similarity/word association. This measure considers not only the connection between the expanded words, obtained from the semantic network, and their co-occurrence with the seed in an external corpus, but also the co-occurrence between the seed and the rest of the (fixed) content words that appear in the poem.

The PMI of a pair of outcomes (e.g. words a and b) quantifies the discrepancy between the probability of their coincidence (e.g. co-occurrence, $P(a,b)$) given their joint distribution and their individual distributions ($P(a)$ and $P(b)$), assuming their independence. Mathematically, PMI can be computed according to equation 1. In order to bound PMI values and thus ease their interpretation, we normalized this measure according to equation 2, as others (Bouma, 2009) did previously.

$$PMI(a,b) = \log \frac{P(a,b)}{P(a) * P(b)} \tag{1}$$

$$NPMI(a,b) = \frac{PMI(a,b)}{-\log(P(a,b))} \tag{2}$$

On the semantic similarity domain, PMI has successfully been used to identify synonyms (Turney, 2001), or to evaluate the coherence of topic models (Newman, Lau, Grieser, and Baldwin, 2010), among others. In both previous cases, it has been noticed that PMI scores had a high correlation with those by human judges. When measured on a large corpus, pairs of words typically used together, and rarely alone (semantically similar), have higher PMI values than those that do not co-occur or only co-occur by accident. In particular, the NPMI between identical words is 1.

For our evaluation, NPMI values were computed based on the Wikipedia editions for each of the three languages. Wikipedia is a large-coverage corpus, available in many languages, and created independently from our grammars and semantic networks. Moreover, it provides a REST API[9] that eases the process of counting the number of documents where a word or a pair of words occur.

The NPMI value between each word and each content word[10] of each poem in the evaluation samples has been computed. These values range from 1, for identical words, to 0, for words that do not co-occur in Wikipedia articles. Some typical values are listed below:

- PT: (*poema*, *poesía*) = 0.660, (*futebol*, *artificial*) = 0.013
- SP: (*procesadores*, *ordenador*) = 0.509, (*construir*, *jugar*) = 0.092
- EN: (*hardware*, *computer*) = 0.484, (*computer*, *flower*) = 0.044

To assess the similarity between each poem and its original seed, we performed an information retrieval task, based on the precision at the top-9 similar poems for each seed. This means that, if all the nine poems generated by a seed are between the nine most similar in the ranking (those with high $NPMI(seed, poem)$), then the precision will be 1. If none of the top-9 poems was generated with the seed, precision will be 0. Table 10 shows the results obtained in this task.

These results show that the words in poems generated by a seed tend to be more similar to this seed than the words in other poems. Therefore, the seeds play a relevant role in the process of generating poems, as it biases the meaning transmitted by them. We can observe that, on average, the highest precision is obtained for English (0.68), and the worst for Spanish (0.53), with Portuguese in the middle (0.66), though with a lower standard deviation than for English (0.19

[9] Check `http://en.wikipedia.org/w/api.php`

[10] Once again, we used the Snowball stopwords.

Seed	**PT**	**ES**	**EN**
1 (*love*)	0.67	0.89	0.78
2 (*artificial*)	0.56	0.00	0.33
3 (*blue*)	0.78	0.00	1.00
4 (*sing*)	1.00	0.44	0.56
5 (*computer*)	0.67	0.33	0.89
6 (*build*)	0.78	0.67	0.44
7 (*football*)	0.33	0.44	0.89
8 (*read*)	0.44	1.00	0.44
9 (*new*)	0.67	0.78	0.89
10 (*poetry*)	0.67	0.78	0.56
Average	0.66±0.19	0.53±0.35	0.68±0.24

Table 10. *Precision at top-9 results, obtained for the evaluation of the semantic similarity between the seeds and the generated poems*

vs 0.24). These differences are explained with the size and nature of the resources used. English is the language with the larger set of renderings, its semantic network is larger than the Spanish one and more systematic. It is also focused on fewer but more common semantic relations (synonymy, hypernymy and part-of) than the Portuguese network, largest and more heterogeneous, with most relations extracted automatically from dictionaries. Last but not least, the English Wikipedia is much larger than the Spanish and the Portuguese (4.9M articles against 1.2M and 890K, respectively), which may contribute to more accurate NPMI values.

For each language, the highest result was 100% precision, but obtained with different seeds. For English, it was *blue*, for Spanish, *leer*, and for Portuguese, *cantar*. The seed that lead to the lower precision was also different for each language. It was *futebol* for Portuguese (P=0.33), *artificial* and *azul* for Spanish (P=0.0), and *artificial* for English (P=0.33). The two seeds with precision 0 for Spanish, *artificial* and *azul*, are adjectives that may modify significantly different nouns and appear in very different contexts, which might have contributed to these low results. Moreover, these seeds are neutral/objective and thus not clearly connected with words with a typical polarity. This results in poems that do not use the seed nor any word with a clear semantic connection with the seed, thus shifting the desired meaning. For instance, for the seed *azul* (blue) and a negative polarity, the seed set is expanded with *pegar* (hit), *mortal* (mortal), *cortar* (cut), and *quitar* (remove). This is also a limitation of the small size of the Spanish semantic network and polarity lexicon.

In general, the higher positions in the rankings are obtained for the poems generated with no target polarity. This was expected because the words obtained with the expansion are the most semantically-relevant for the original seed. Still, while this behavior was clear for Portuguese and Spanish, for English there was not a clear order in the poems generated with different target polarities. In particular, for Portuguese there were four seeds (*amor*, *azul*, *futebol*, *poesía*) where the top-3 similar poems were those generated without polarity. Though different from the Portuguese seeds, for Spanish there were also four seeds (*amor*, *cantar*, *ordenador*, *poesía*) where this happen, but for English there was only one (*computer*).

6 Final discussion

PoeTryMe and its architecture constitute an effort to establish a set of procedural modules that might be applied to solve the same task across different languages. The procedures implemented in these modules are specific to a particular approach to poetry generation, which involves the automated extraction of line templates based on semantic relations between pairs of keywords, and completion of those templates with words that are different but analogously related on a one-to-one basis. For this particular approach to poetry generation, the PoeTryMe architecture isolates the specific components that are language-dependent, and provides different instantiations of these components for Portuguese, Spanish and English. The establishment of this differentiation between the components that are language specific and those that are language independent is an important contribution of this paper.

Although an effort has been made to consider several European languages, the distinction between language-dependent and language-independent parts of the system may change if radically different languages are considered. This is to be expected, and no claim is intended as to the general applicability of the distinction as presented here. Nevertheless, it is important to note that we have isolated a common core that can be applied across more than one language, allowing the same operational principle to be tested across languages, and providing some degree of comparability across those languages, which was not available for prior systems. With the proposed setting, and providing that the required knowledge resources are available – semantic network, collection of poetry, morphology lexicon, polarity lexicon, and a tool for syllable-related operations – , instantiating PoeTryMe in additional languages should be a matter of identifying these resources and plugging them in, possibly after performing minor adaptations (e.g. in data format).

The evaluation procedure was designed with its application to the three language-specific instances of PoeTryMe in mind and targeted three aspects of poetry generation that the authors considered as important contributions to the added value of the results. Conformance to a given poetic form is a basic requirement for poetry generation when understood in a classical sense. If the system produces output that does not meet the required constraints on form, it is less likely to be seen as producing poetry. The variation across the output itself is considered a measure of the richness of the generative procedures. Output that is repetitive is a signal of poor generative procedures or of scarcity in the knowledge resources employed to feed them. The suitability of the outputs with respect to a given input query, in the case of PoeTryMe a set of seed words, provides an indication of the ability that the system has to explore particular regions of the conceptual space of possible outputs, and to come up with solutions that match a particular request expressed by the user.

Although the presented evaluation is not necessarily complete, the combination of the three target aspects covers features that are valuable to all poetry generation efforts, irrespective of the particular generative approach that is being applied in each case. Given that it is computed automatically using established measures, and thus do not rely on subjective human opinions, the applied procedures might be applicable as an evaluation template to other poetry systems to provide a benchmark of comparability.

In our case, we have confirmed that the proposed solution can indeed produce poetry in the three target languages, successfully enough, at least in the assessed aspects: poems revealed to have a regular metre, frequent rhymes, to exhibit an interesting degree of variation, and to be semantically associated with the initially given seeds. Differences on the results for each language were commented, but most of them are caused by the different sizes and qualities of the underlying knowledge resources. They should thus be seen as a comparison between the three language instantiations of PoeTryMe, rather than a comparison between the three languages. So far, our main concerns were to identify and integrate resources with a similar organization that would meet the minimum requirements for language-specific instantiations. The used semantic networks have different sizes (distinct words, triplets), relation types and average degrees. The text collections exploited for discovering the generation grammars also had a different number of documents and effectively useful lines. The morphology and polarity lexicons should have also played their role, but it is not easy to measure the impact of each of the previous differences clearly.

Focusing more on the languages and less on each instantiation would require a comparable version of the resources, which is not straightforward to achieve. The semantic networks would have to have the same size, to cover the same relation types, and to be a translation of each other or, at least, to be focused on the same domains. The generation grammars would have to be created manually, following the same guidelines, or extracted with the aforementioned semantic networks from comparable collections of text, either a parallel corpus or, at least, poems and other documents with similar forms and on the same topics. This means that the languages for which resources are richer would have to see their resources simplified according to those for which resources are poorer. Resulting poems would probably be less genuine as well.

We should add that, in order to focus on its multilingual extension, we limited the tested parameters of PoeTryMe to a single form of poetry (sonnet) while varying the seeds and the target polarity. Other parameters will be tested in the future, and some already have in other instantiations of PoeTryMe that confirm the flexibility of its architecture. Besides the generation of other poetry forms, PoeTryMe has been adapted to other purposes, such as poetry inspired by tweets (Gonçalo Oliveira, 2016), concept maps extracted from text (Gonçalo Oliveira and Oliveira Alves, 2016), or the generation of song lyrics (Gonçalo Oliveira, 2015), where rhythm plays an even more relevant role.

The search for correlations between automatic evaluation and human evaluation is a promising avenue for further research. However, in undertaking such a task extreme care must be taken to avoid the pitfalls of undesirable bias in evaluators arising from preconceptions of what is to be expected from computer generated poetry. Not to mention that it might be challenging to find a comparable group of judges for each language.

A simplified version of PoeTryMe is available in the TryMe section of `http://poetryme.dei.uc.pt/`. Given a list of seed words and a surprise factor, this version enables the generation of poetry in the three covered languages, structured according to a set of available poetry forms.

Acknowledgements

This work was supported by projects PROSECCO and ConCreTe.

Part of this work was developed during short term visits funded by the PROSECCO CSA project, European 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

Aamodt, A. and E. Plaza (1994). Case-based reasoning; foundational issues, methodological variations, and system approaches. *AI COMMUNICATIONS 7*(1), 39–59.

Agirrezabal, M., B. Arrieta, A. Astigarraga, and M. Hulden (2013, August). Pos-tag based poetry generation with wordnet. In *Proceedings of the 14th European Workshop on Natural Language Generation*, Sofia, Bulgaria, pp. 162–166. ACL Press.

Barbieri, G., F. Pachet, P. Roy, and M. D. Esposti (2012). Markov constraints for generating lyrics with style. In *Proceedings of 20th European Conference on Artificial Intelligence (ECAI)*, Volume 242 of *Frontiers in Artificial Intelligence and Applications*, pp. 115–120. IOS Press.

Bouma, G. (2009). Normalized (pointwise) mutual information in collocation extraction. In *Proceedings of the Biennial GSCL Conference*, pp. 31–40. Gunter Narr Verlag.

Brin, S. and L. Page (1998). The anatomy of a large-scale hypertextual web search engine. *Computer Networks 30*(1-7), 107–117.

Carolyn E. Lamb, D. G. B. and C. L. Clarke (2015). Can human assistance improve a computational poet? In K. Delp, C. S. Kaplan, D. McKenna, and R. Sarhangi (Eds.), *Proceedings of Bridges 2015: Mathematics, Music, Art, Architecture, Culture*, Phoenix, Arizona, pp. 37–44. Tessellations Publishing.

Charnley, J., S. Colton, and M. T. Llano (2014, June). The flowr framework: Automated flowchart construction, optimisation and alteration for creative systems. In *Proceedings of 5th International Conference on Computational Creativity, ICCC 2014*, Ljubljana, Slovenia.

Charnley, J., A. Pease, and S. Colton (2012, May). On the notion of framing in computational creativity. In *Proceedings of the 3rd International Conference on Computational Creativity*, ICCC 2012, Dublin, Ireland, pp. 77–81.

Church, K. W. and P. Hanks (1990, mar). Word association norms, mutual information, and lexicography. *Computational Linguistics 16*(1), 22–29.

Colton, S., J. Goodwin, and T. Veale (2012). Full FACE poetry generation. In *Proceedings of 3rd International Conference on Computational Creativity*, ICCC 2012, Dublin, Ireland, pp. 95–102.

Colton, S. and G. A. Wiggins (2012). Computational creativity: The final frontier? In *Proceedings of 20th European Conference on Artificial Intelligence (ECAI 2012)*, Volume 242 of *Frontiers in Artificial Intelligence and Applications*, Montpellier, France, pp. 21–26. IOS Press.

Das, A. and B. Gambäck (2014, June). Poetic machine: Computational creativity for automatic poetry generation in bengali. In *Proceedings of 5th International Conference on Computational Creativity*, ICCC 2014, Ljubljana, Slovenia.

Fellbaum, C. (Ed.) (1998). *WordNet: An Electronic Lexical Database (Language, Speech, and Communication)*. The MIT Press.

Gervás, P. (2000). 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*, Birmingham, UK, pp. 93–100.

Gervás, P. (2001). An expert system for the composition of formal Spanish poetry. *Journal of Knowledge-Based Systems 14*(3–4), 181–188.

Gervás, P. (2013a). Computational modelling of poetry generation. In *Proceedings of the AISB'13 Symposium on Artificial Intelligence and Poetry*.

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

Gervás, P. (2000). A logic programming application for the analysis of Spanish verse. In *1st International Conference on Computational Logic*, Imperial College, London, UK, pp. 1330–1344.

Gonçalo Oliveira, H. (2012, August). PoeTryMe: a versatile platform for poetry generation. In *Proceedings of the ECAI 2012 Workshop on Computational Creativity, Concept Invention, and General Intelligence*, C3GI 2012, Montpellier, France.

Gonçalo Oliveira, H., L. Antón Pérez, H. Costa, and P. Gomes (2011, December). Uma rede léxico-semântica de grandes dimensões para o português, extraída a partir de dicionários electrónicos. *Linguamática 3*(2), 23–38.

Gonçalo Oliveira, H. and A. Cardoso (2015). Poetry generation with PoeTryMe. In T. R. Besold, M. Schorlemmer, and A. Smaill (Eds.), *Computational Creativity Research: Towards Creative Machines*, Atlantis Thinking Machines, Chapter 12, pp. 243–266. Atlantis-Springer.

Gonçalo Oliveira, H., F. A. Cardoso, and F. C. Pereira (2007). Tra-la-Lyrics: an approach to generate text based on rhythm. In *Proceedings of 4th International Joint Workshop on Computational Creativity*, London, UK, pp. 47–55. IJWCC 2007.

Gonçalo Oliveira, H., R. Hervás, A. Díaz, and P. Gervás (2014, June). Adapting a generic platform for poetry generation to produce spanish poems. In *Proceedings of 5th International Conference on Computational Creativity*, ICCC 2014, Ljubljana, Slovenia.

Gonzalez-Agirre, A., E. Laparra, and G. Rigau (2012, LREC'12). Multilingual central repository version 3.0. In *Proceedings of the 8th International Conference on Language Resources and Evaluation*, pp. 2525–2529. ELRA.

Gonçalo Oliveira, H. (2015, December). Tra-la-lyrics 2.0: Automatic generation of song lyrics on a semantic domain. *Journal of Artificial General Intelligence 6*(1), 87–110. Special Issue: Computational Creativity, Concept Invention, and General Intelligence.

Gonçalo Oliveira, H. (2016). Automatic generation of poetry inspired by twitter trends. In *Knowledge Discovery, Knowledge Engineering and Knowledge Management (Post-conference Proceedings of IC3K — Revised Selected Papers)*, Volume 631 of *CCIS*, pp. 13–27. Springer.

Gonçalo Oliveira, H. and A. Oliveira Alves (2016). Poetry from concept maps – yet another adaptation of PoeTryMe's flexible architecture. In *Proceedings of 7th International Conference on Computational Creativity*, ICCC 2016, Paris, France.

Jordanous, A. (2012). A standardised procedure for evaluating creative systems: Computational creativity evaluation based on what it is to be creative. *Cognitive Computation 4*(3), 246–279.

Lamb, C., D. Brown, and C. Clarke (2016). How digital poetry experts evaluate digital poetry. In *Proceedings of 7th International Conference on Computational Creativity*, ICCC 2016, Paris, France.

Lin, C.-Y. and F. J. Och (2004). Automatic evaluation of machine translation quality using longest common subsequence and skip-bigram statistics. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, ACL 2004. ACL Press.

Liu, B., M. Hu, and J. Cheng (2005). Opinion observer: Analyzing and comparing opinions on the web. In *Proceedings of the 14th International Conference on World Wide Web*, WWW '05, New York, NY, USA, pp. 342–351. ACM.

Malmi, E., P. Takala, H. Toivonen, T. Raiko, and A. Gionis (2016). Dopelearning: A computational approach to rap lyrics generation. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, pp. 195–204.

Manurung, H. (1999). A chart generator for rhythm patterned text. In *Proceedings of 1st International Workshop on Literature in Cognition and Computer.*

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

Misztal, J. and B. Indurkhya (2014, 06/2014). Poetry generation system with an emotional personality. In *5th International Conference on Computational Creativity, ICCC 2014*, Ljubljana, Slovenia.

Netzer, Y., D. Gabay, Y. Goldberg, and M. Elhadad (2009). Gaiku: generating haiku with word associations norms. In *Proceedings of the Workshop on Computational Approaches to Linguistic Creativity*, CALC'09, pp. 32–39. ACL Press.

Newman, D., J. H. Lau, K. Grieser, and T. Baldwin (2010). Automatic evaluation of topic coherence. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10, pp. 100–108. ACL Press.

Oulipo, A. (1981). *Atlas de littérature potentielle.* Number vol. 1 in Collection Idées. Gallimard.

Padró, L. and E. Stanilovsky (2012, May). Freeling 3.0: Towards wider multilinguality. In *Proceedings of the Language Resources and Evaluation Conference*, LREC'12, Istanbul, Turkey. ELRA.

Pease, A. and S. Colton (2011). On impact and evaluation in computational creativity: A discussion of the turing test and an alternative proposal. In *Proceedings of the 3rd AISB symposium on AI and Philosophy, University of York, York, UK.*

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

Ramakrishnan A, A., S. Kuppan, and S. L. Devi (2009). Automatic generation of Tamil lyrics for melodies. In *Proceedings of the Workshop on Computational Approaches to Linguistic Creativity*, CALC'09, pp. 40–46. ACL Press.

Ranchhod, E., C. Mota, and J. Baptista (1999). A computational lexicon of portuguese for automatic text parsing. In *Proceedings of SIGLEX99: Standardizing Lexical Resources – 37th Annual Meeting of the ACL*, College Park, MD, USA, pp. 74–80. ACL Press.

Rashel, F. and R. Manurung (2014, June). Pemuisi: A constraint satisfaction-based generator of topical indonesian poetry. In *Proceedings of 5th International Conference on Computational Creativity*, ICCC 2014, Ljubljana, Slovenia.

Reiter, E. and R. Dale (2000). *Building natural language generation systems.* New York, NY, USA: Cambridge University Press.

Santos, D. and E. Bick (2000). Providing Internet access to Portuguese corpora: the AC/DC project. In *Procs. of 2nd Intl. Conf. on Language Resources and Evaluation*, LREC 2000, pp. 205–210.

Silva, M. J., P. Carvalho, and L. Sarmento (2012). Building a sentiment lexicon for social judgement mining. In *Proceedings of Computational Processing of the Portuguese Language - 10th International Conference (PROPOR 2012)*, Volume 7243 of *LNCS*, Coimbra, Portugal, pp. 218–228. Springer.

Tobing, B. C. L. and R. Manurung (2015, Jun). A chart generation system for topical metrical poetry. In *Proceedings of the 6th International Conference on Computational Creativity, Park City, Utah, USA*, ICCC 2015, Park City, Utah, USA.

Toivanen, J. M., O. Gross, and H. Toivonen (2014, 06/2014). The officer is taller than you, who race yourself! using document specific word associations in poetry generation. In *5th International Conference on Computational Creativity, ICCC 2014*, Ljubljana, Slovenia.

Toivanen, J. M., M. Järvisalo, and H. Toivonen (2013). Harnessing constraint programming for poetry composition. In *Proceedings of the 4th International Conference on Computational Creativity*, ICCC 2013, pp. 160–167. The University of Sydney.

Toivanen, J. M., H. Toivonen, A. Valitutti, and O. Gross (2012, May). Corpus-based generation of content and form in poetry. In *Proceedings of the 3rd International Conference on Computational Creativity*, ICCC 2012, Dublin, Ireland, pp. 211–215.

Turney, P. D. (2001). Mining the web for synonyms: PMI–IR versus LSA on TOEFL. In *Proceedings of 12th European Conference on Machine Learning, ECML 2001*, Volume 2167 of *LNCS*, pp. 491–502. Springer.

Urizar, X. S. and I. S. V. Roncal (2013). Elhuyar at tass 2013. In *Proceedings of XXIX Congreso de la Sociedad Española de Procesamiento de lenguaje natural. Workshop on Sentiment Analysis at SEPLN (TASS2013)*, Madrid, pp. 143–150.

Valitutti, A., A. Doucet, J. Toivanen, and H. Toivonen (2016, 5). Computational generation and dissection of lexical replacement humor. *Natural Language Engineering 22*(5), 1–23.

van der Velde, F., R. A. Wolf, M. Schmettow, and D. S. Nazareth (2015). A semantic map for evaluating creativity. In *Proceedings of the 6th International Conference on Computational Creativity June*, pp. 94.

Veale, T. (2012). *Exploding The Creativity Myth: The Computational Foundations of Linguistic Creativity*. Bloomsbury Publishing.

Wong, M. T. and A. H. W. Chun (2008). Automatic haiku generation using VSM. In *Proceeding of 7th WSEAS International Conference on Applied Computer & Applied Computational Science*, ACACOS '08, Hangzhou, China.

Yan, R., H. Jiang, M. Lapata, S.-D. Lin, X. Lv, and X. Li (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, pp. 2197–2203. AAAI Press.

Zhang, X. and M. Lapata (2014). Chinese poetry generation with recurrent neural networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar*, pp. 670–680. ACL Press.