

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/228609235>

# Wasp: Evaluation of different strategies for the automatic generation of spanish verse

## Article

---

CITATIONS

42

---

READS

62

## 1 author:



**Pablo Gervás**

Complutense University of Madrid

201 PUBLICATIONS 1,214 CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:



IDIlyCo: Digital Inclusion, Language and Communication [View project](#)

All content following this page was uploaded by [Pablo Gervás](#) on 13 December 2013.

The user has requested enhancement of the downloaded file. All in-text references [underlined in blue](#) are added to the original document and are linked to publications on ResearchGate, letting you access and read them immediately.

# WASP: Evaluation of Different Strategies for the Automatic Generation of Spanish Verse

Pablo Gervás

Departamento de Inteligencia Artificial  
Escuela Superior de Informática  
Universidad Europea CEES  
28670 Villaviciosa de Odón  
Spain  
pg2@dinar.esi.uem.es

## Abstract

WASP is a forward reasoning rule-based system that takes as input data a set of words and a set of verse patterns and returns a set of verses. Using a generate and test method, guided by a set of construction heuristics obtained from formal literature on Spanish poetry, the system can operate in two modes: either generating an unrestricted set of verses, or generating a poem according to one of three predefined structures (*romance*, *cuarteto*, or *terceto*). Five different construction heuristics are tested over different combinations of two sets of initial data, one obtained from a classic poem and one obtained from a paragraph of a doctoral thesis in linguistics. A set of numerical parameters are extracted from each test, and evaluated in search of significant correlations. The aim is to ascertain the relative importance of size of initial vocabulary, choice of words, choice of verse patterns and construction heuristics with respect to the general acceptability of the resulting verse.

## 1 Programs that Write Poetry Automatically

The creation of programs that write poetry automatically has been a recurring dream within the AI community, but it has always been assigned a very low priority. Practical applications in the area of natural language processing, such as natural language database interfaces, information retrieval and extraction, automatic translation, and dialogue systems provide more immediate rewards.

On one hand the automatic generation of poetry involves advanced linguistic skills and common sense, two of the major challenges that face AI in general. On the other hand it involves an important amount of creativity and sensibility. These ingredients are very difficult to characterise formally, and very little is known about how they might be treated algorithmically.

On the positive side, poetry has the advantage of not requiring exaggerate precision. If one accepts that the main aim of a poem is to be pleasing rather than conveying a meaningful message, the general problem becomes tractable. The present paper considers how the different parameters that can be controlled by the generating program affect the acceptability of the result. The set of parameters to be monitored are: size of initial vocabulary, choice of words, choice of verse patterns, and construction heuristics. The elusive concept of acceptability of a verse is determined by resorting to hand evaluation by

a team of volunteers. By searching for correlations between the strategy and initial data used to generate a verse and the positive or negative evaluation of the resulting verse, information is obtained about the relative relevance of these parameters to the end result.

### 1.1 Guiding Heuristically the Random Generation of a Verse

Poems written by combining randomly a given set of words rate very poorly with discerning readers. For the words to make sense together, they must be organised according to particular patterns. A possible course of action would be to provide the system with adequately rich lexicon, syntax and semantics for the language involved. Results obtained with inadequate formalisms are too rigid and tend to have a mechanical ring to them. The system presented in this paper resorts to a radical simplification of the underlying linguistic skills. The exhaustive knowledge approach is abandoned in favour of a heuristic engineering solution. Only the barest outline of a grammatical outline is provided (in the form of a verse pattern) to ensure syntactic correctness. Semantic correctness is not enforced, on the understanding that creativity in poetry relies to a certain extent on daring transgressions (such as imaginative metaphors). The aim of the paper is to establish whether acceptable verse may be obtained by controlling other parameters within these initial restrictions. The hope is

to identify whether the elementary ingredients considered can be manipulated smartly enough to produce a pleasing phrase.

## 1.2 The Effect of the Selection of Initial Data

Under these restrictions, Spanish has been chosen as a test language. The phonetics of Spanish are quite straightforward to obtain from the written word. Most letters in Spanish sound the same wherever they appear in a piece of text, so the metrics, or the syllabic division, of a verse can be worked out algorithmically (RAE, 1986). Spanish scholars have a love for rules, and there is a good set of formal rules (Quilis, 1985) describing the conditions that a poem must fulfil in order to be acceptable.

Given such a set of rules, the challenge becomes a simple problem of transforming the given evaluation rules (designed to be applied to an existing poem in order to ascertain its acceptability) into the corresponding construction rules. These rules have to be applied to an initial set of data consisting of:

- a given vocabulary (given a set of words, the poet will choose only some of them, this process of selection must surely play a role in the quality of the final result), and
- a particular choice of ways of combining the chosen words (word order, frequency of adjectives, length of verse...) represented as a set of verse patterns.

The selected vocabulary is a set of words that includes extra information about part of speech roles, number of syllables of each word, position of stressed syllables, and rhyme. The system cannot handle morphological variations, so it considers the singular and plural, masculine and feminine forms of a word as totally distinct (and different tenses of a verb also). This decision reduces the complexity of the generation process to pattern matching between word categories and verse patterns, but it has consequences on the quality of the resulting verses.

The set of valid words is stored as facts of the form:

```
(word (cual luz)
      (numsil 1)
      (acento 1)
      (emp 0)
      (term 0)
      (cat susfem)
      (rima uz))
```

where:

- the field `cual` is the word itself, to be used as key when retrieving the rest of the information
- the field `numsil` shows the number of syllables of the word
- the field `acento` shows the position of the stressed syllable from the beginning of the word

- the field `emp` indicates whether the word starts with a vowel (required to scan a verse)
- the field `term` indicates whether the word ends with a vowel
- the field `rima` contains the rhyme of the word

Each verse pattern is a list of tags, and each tag acts as place keeper for a possible word of the verse. The tag is actually a string that represents information about part of speech, number, and gender of the word that would stand in that particular place in the pattern.

This information is stored as facts of the form:

```
(patron prep artmas adjmas susmas adjmas adjmas)
```

where `patron` is the generic fact name and the following items are tags for the categories of the words in a particular verse.

Patterns act as seed for verses, therefore a pattern determines the number of words in a verse, the particular fragment of sentence that makes up the verse, and the set of words that can be considered as candidates for the verse. By following this heuristic shortcut, WASP is able to generate verses with no knowledge about grammar or meaning.

## 1.3 Comparing Different Strategies

This paper presents the results of several experiments designed to determine the relative merits of different strategies for generating verses. These strategies are considered with respect to several parameters.

A first group of strategies plays a role in determining simply the number of verses generated from a given set of initial data. These strategies can be tested by generating simple lists of independent verses and evaluating the acceptability of the results. The analysis of the results should determine optimal choices for: (1) the method used to avoid having repeated words in a given verse or poem, and (2) the specific definition used to validate each successive draft of a verse.

A second group of strategies is expected to affect the quality of complete poems generated by the system. These strategies are tested by attempting to generate a number of poems for each strophic form and evaluating the results. Conclusions should provide information about: (1) effect of length of vocabulary, (2) effect of number of patterns, (3) effect of 'informed' assignment of verse borders (4) effect of interaction between patterns and words in the vocabulary (5) effect of extending the words of the vocabulary beyond those present in the initial data

## 2 A Brief Introduction to Spanish Poetry

This section outlines a few concepts related to Spanish poetry and metric that play a role in the definition of the

WASP system. For more extensive treatment, see Quilis (1985). A good summary in English is available at Williamsen (2000).

## 2.1 When and Why a Verse is Valid

Formal analysis of poetry considers the position of stressed syllables over a verse. For the verse to sound pleasing, the prosodic accents must be distributed according to precise patterns. This distribution of prosodic patterns provides the quality of being pleasant to the ear.

For instance, for an eleven syllable long verse to sound pleasing, it needs some of the stressed syllables of its words to fall on certain specific positions. It is not necessary for the stressed syllables of every word in the verse to be in specific positions. It is enough for certain strategic syllabic positions within the verse to have a stressed syllable. The literature (Quilis, 1985) requires stressed syllables to fall either on positions 1, 6 and 10; 2, 6 and 10; 3, 6 and 10; 4, 6 and 10; or 4, 8 and 10. In the following examples, stressed syllables falling in key positions are underlined:

en <u>ver</u> des ho <u>ja</u> s <u>ví</u> que se to <u>rn</u> a <u>ba</u> n	2, 6, 10
nunca <u>fue</u> co <u>ra</u> zo <u>n</u> ; si pre <u>gun</u> tado	3, 6, 10
De tan he <u>rm</u> oso <u>fue</u> go co <u>nu</u> mido	4, 6, 10
soy lo de <u>má</u> s, en lo de <u>má</u> s soy <u>mud</u> o	4, 8, 10

## 2.2 Synaloepha: Counting the Number of Syllables

A metric syllable does not always match the corresponding morphological syllable. When a word ends in a vowel and the following word starts with a vowel, the last syllable of the first word and the first syllable of the following word constitute a single syllable. This is known as *synaloepha* (see Quilis (1985), or Williamsen (2000) for an overview in English), and it is one of the problems that we are facing. For instance, the following verse

bástete amor lo que ha por mí pasado  
13 syllables

turns into:

bástete - amor lo **que** - **ha** por mí pasado  
11 syllables

because it shows two instances of synaloepha (marked in bold).

## 2.3 Building Poems

A *poem* may be an unstructured sequence of verses, but this paper is concerned specifically with poems that make use of known *strophic forms* or *stanzas* (particular patterns of structuring the verses of a poem according to

verse length, metre, and rhyme). In such cases, the formal rules that govern the chosen strophic form can be used to guide the generation process.

A poem may consist of a single stanza or several stanzas together (in which case the different stanzas are usually separated by an empty line).

For the present purposes, only three of the simplest strophic forms need be considered:

1. *romances*, a stanza of several verses where all even numbered verses rhyme together and the rhyme of odd verses is free
2. *cuartetos*, a stanza of four verses where the two outer verses rhyme together and the two inner verses rhyme together
3. *tercetos encadenados*, a longer poem made up of stanzas of three verses linked together by their rhyme in a simple chain pattern ABA BCB CDC...

The following are elementary examples of these stanzas:

### Type 1: Romance

Por el Val de las Estacas  
pasó el Cid a mediodía.  
En su caballo Babieca  
muy gruesa lanza traía.  
Va buscando al moro Abdala  
que enojado le tenía.  
...

The rhyme of each verse is marked in bold.

### Type 2: Cuarteto

Muérome por llamar Juanilla a **Juana**,  
que son de tierno amor afectos **vivos**,  
y la cruel, con ojos fugitivos,  
hace papel de yegua galiciana.

The different rhymes of each verse are marked in bold and italic.

### Type 3: Tercetos encadenados

Alma a quien todo un dios prisión ha **sido**,  
venas que humor a tanto fuego han **dado**,  
medulas que han gloriosamente **ardido**  
su cuerpo dejará, no su **cuidado**;  
serán ceniza, mas tendrán **sentido**;  
polvo serán, mas polvo **enamorado**.

These three types have been chosen because each shows a different structural characteristic that may affect the overall result. Type 1 presents a recurring rhyme that flows all along the poem. It uses only one rhyme, so many words that rhyme together are required for an acceptable result (our starting data have proven to be poor choices in this respect). Type 2 presents a very simple but rigid

structure. It stands for the simplest possible stanza with enough complexity to be distinguishable from prose (the simplification employed with respect to syntax/semantics makes it difficult for shorter poems to sound acceptable). Type 3 presents a simple structure that recurs throughout the poem, but with the rhyme changing slowly as it moves down.

### 3 System Description

WASP (Wishful Automatic Spanish Poet) is a forward reasoning rule-based system that takes as input data a set of words and a set of verse patterns and returns a set of verses. Since the aim of the experiment is to compare different methods of generation, WASP is in fact a set of programs, each one applying a different strategy to generate verses. The different programs that integrate WASP are written in CLIPS (Riley, 2000), a rule-based system shell developed by NASA.

WASP operates over a data structure defined as the *draft of the current verse*. This structure is a list of words that is built incrementally. The algorithms used by WASP follow a generate and test pattern. At each stage of the generation process the draft is tested to ensure that the metric conditions are being met. The moment conditions are violated, the draft of the current verse is rejected and the system starts a draft for a new verse.

#### 3.1 Initial Data

The system requires a set of initial data to start the generation process: a vocabulary and a set of patterns. The choice of vocabulary greatly determines the tense and the topic of the poem. The set of verse patterns can be considered as a set of descriptions of past cases, in the sense that it encodes information about important parameters (number of words per verse, rate of adjectives per noun, tense...) while allowing a certain leeway in terms of specific content (particular words) of the present solution.

The set of initial data is obtained as follows. Given a block of text, it is split into fragments of shorter length. All the words in the poem are included in the vocabulary. The resulting fragments of the original text are used to produce the reference patterns. This is done by checking each word of the fragment in the vocabulary and substituting for it the corresponding category.

When an actual poem is used as the original block of text, the existing division into verses can be used. Alternative divisions can be used to test whether this particular decision of the poet plays a significant role in the quality of the results.

In order to compare the effect of the choice of vocabulary and the choice of verse pattern, two distinct sets of data are used to test the programs. The first set of data is obtained from a classic Spanish poem, a Sixteenth Century sonnet by Garcilaso de la Vega (Navarro, 1973). The second set of data is taken randomly from an academic

work in the field of linguistics. A certain paragraph (Rico, 1994) of equivalent size is chosen, all the words in the paragraph are included, and a set of reference patterns is built by splitting the paragraph into chunks of roughly the required size and encoding the necessary information.

#### 3.2 Basic Algorithm

Generation starts with the selection of an appropriate verse pattern, based on criteria designed to ensure that there is a minimum of coherence across verse boundaries. From this pattern an empty draft of the current verse is generated.

The elementary generation cycle can be described as follows:

1. randomly choose from the given vocabulary a word that matches the first category of the current verse pattern
2. append it to the draft of the current verse
3. eliminate the corresponding category from the current verse pattern
4. test whether the resulting verse draft satisfies the conditions of the strategy being used – and the required length of verse in syllables
5. if the conditions are satisfied, iterate from 1.
6. verses that either violate the conditions, or overshoot or fall short of the given number of syllables are rejected

#### 3.3 Strategies for Single Verse Generation

In order to appreciate more clearly the effect on single verse generation, the relevant experiments are carried out without the additional restrictions that determine poem generation. In practical terms, this means that the verse pattern used for each verse is chosen at random from the initial data.

##### 3.3.1 Avoiding Word Repetition

Three different possibilities are considered:

1. simple random combination of the given words into the given patterns
2. ensure no word appears twice in the same poem by eliminating words that have already been used
3. ensure no word appears twice in the same poem by noting down words that have already been used (this allows a rough procedure of garbage collection to avoid losing words used in failed attempts)

Strategy 2 simply deletes used words from the set of available data. If the current draft is subsequently rejected, the words used so far are lost and cannot be used elsewhere.

Strategy 3 makes a note of words that have been used in the current draft and returns them to the set of available data if the draft is rejected.

### 3.3.2 Validating Successive Drafts of a Verse

Three different possibilities are considered:

1. simply test that the number of syllables is yet smaller than the required length
2. make sure stressed syllables of the words chosen at each step fall into the positions deemed acceptable by the formal rules
3. implement the previous strategy (make sure stressed syllables of the words chosen at each step fall into the positions deemed acceptable by the formal rules) but taking into account the possibility of synaloepha occurring between words

WASP aims for a generic verse length of eleven syllables. For strategies 2 and 3 that apply formal rules for acceptability in terms of position of stressed syllables, we require that the stressed syllable of any word added to a partially completed verse falls either on positions 1, 2, 4, 6, 8 or 10.

## 3.4 General Strategy for Generating Poems

The generation of poems requires two additional issues to be solved, both related to the restrictions imposed on each verse by the previous verses of the poem. One concerns the choice of verse pattern to use for the next verse. This issue is independent of the particular strophic form sought. The other concerns the rhyme to use for the next verse, and is governed by the particular rules of each strophic form.

### 3.4.1 Selecting Verse Pattern for the Next Verse

The selection of a verse pattern for the next verse of a poem must take into account the need for coherence between verses across verse boundaries. When operating in the complete poem mode, WASP stores all the verses generated so far, numbered according to their order in the poem. The verse pattern for the next verse is chosen according to the following criteria: the first word category of the selected pattern must occur in some verse pattern of the initial data immediately after the last word category of the previous verse. For instance, if the pattern used for the previous verse was (patron ... adjmas), any verse pattern of the form (patron susmas...) is acceptable provided there is a third verse pattern of the form (patron ... adjmas susmas...).

Because verse patterns are produced with no discrimination between patterns that correspond to beginning, end, or middle sections of a sentence, the system must allow the selection of a verse pattern at random if this condition is not met. This solution is an important source of errors in the final results, and it is an obvious candidate for refinement in subsequent versions of the system.

### 3.4.2 Selecting Rhyme for the Next Verse

The selection of a rhyme for the next verse imposes an additional restriction on the set of verse patterns that are acceptable candidates for continuing a given poem. As well as fulfilling the condition outlined above, the verse pattern for the next verse must end in a word category for which there is a word in the vocabulary with the required rhyme. The simplest method of achieving this is to select the required word and then find a verse pattern that matches the restrictions imposed on its initial word category (by the previous verse) and on its final word (already chosen at this stage).

The system is designed to generate poems using three different strophic forms: *romances*, *cuartetos*, and *terceros encadenados*.

For all three cases, the first rhymes are fixed by two initial random choices (selection of a verse pattern to start with, and selection of an end word for that particular pattern). This is because there is no initial reference. Once the first verses have been established, WASP ensures that the following verses fit the corresponding stanza (if the starting data and the verse construction strategies will allow it).

In the case of *romances* and *terceros encadenados* the conditions are formulated using a modulo operation on the verse number, therefore the system has the theoretical potential of generating poems of any length of verses. The actual restrictions faced by the system are imposed by the initial vocabulary (both by the number of rhymes included and by the sheer number of words, since no word repetition is accepted within the same poem).

For *cuartetos*, the system can only generate a single stanza of four verses.

## 4 Evaluation of Results

Three different sets of experiments were carried out.

In each experiment of the first set, the versions of the system corresponding to different strategies for avoiding word repetition were compared.

The experiments of the second set were designed to evaluate which of the strategies for validating the current draft of a verse gave better results.

For both the first and the second sets of experiments, each competing version of the system attempted to generate a thousand verses, operating in single verse mode. A classic Spanish poem was used to provide initial data, and division of the poem into verses was respected.

The third set of experiments was carried out using only a version of the system that combined the strategies that had obtained better results over the previous sets. Comparisons were established between results obtained for different combinations of initial data. In this set, each competing version attempted to generate twelve poems for each one of the possible strophic forms.

#### 4.1 Avoiding Repetition over Single Verses

Table 1 shows the average percentage results for each strategy. In each case, the program was allowed to carry out 1000 iterations, using a generate and test method. For this part of the evaluation, verse correctness was evaluated automatically using a logic programming application for the analysis of Spanish verse (Gervás, 2000). The evaluating application applies strictly the formal rules found in the literature, and validates a verse if it fulfils the required conditions. As such, it constitutes an impartial judge of the correctness of each verse.

The following strategies are compared:

1. simple random combination
2. eliminate used words
3. annotate and replace used words

Table 1: Avoiding Verse Repetition

Version	% Generated	% Correct	% Corr. Gen.
1	35.50	12.90	36.34
2	0.30	0.20	66.67
3	38.40	15.40	40.10

The results for strategy 1 show that, under the minimum of restrictions, only 35 % of the attempts actually generate a verse. This implies that, if strategy 2 is applied, 75 % of the times words are being used up in vain. This matches up with the observed results, where generation drops drastically after the first few attempts. Strategy 3, providing a reasonable solution to word repetition, improves the results of strategy 1 both in terms of number of verses generated altogether and in terms of number of correct verses generated.

For the rest of the experiments, strategy 3 is used.

#### 4.2 Validating Current Draft of a Verse

Table 2 shows results to the second set of experiments. This set was carried out in a similar manner as the above, and evaluated in the same way.

The strategies compared in this case were:

1. simply count the number of syllables
2. count number of syllables and check position of stressed syllables
3. count number of syllables and check position of stressed syllables taking synaloepha into account

3. count number of syllables and check position of stressed syllables taking synaloepha into account

Table 2: Validating Verse Draft

Version	% Generated	% Correct	% Corr. Gen.
3	38.40	15.40	40.10
4	38.90	18.80	48.33
5	46.60	33.60	72.10

Strategy 3 referred to in this table is actually the same as strategy 3 of table 1, and results for it are given again only for ease of reference. The table shows that imposing additional restrictions on the validation of the current draft does not result in a smaller number of verses being generated. In fact, the number of verses generated increases steadily as more restrictions are applied. Furthermore, the increase is noticeable greater when the complete set of restrictions is applied. This can be attributed to the fact that the initial data correspond to a poem that actually fulfils these conditions strictly. The given vocabulary, and the given patterns perform optimally for the complete set of restrictions.

For the rest of the experiments, strategy 5 is used.

#### 4.3 Poem Generation and Initial Data

Exhaustive tests were carried for different combinations of the initial data, using generating strategy 5 on all cases.

The following parameters were combined:

- different sources for initial data (poetic or academic text)
- extensions of the vocabulary beyond the original text
- allowing repetitions of words other than nouns, adjectives, adverbs or verbs
- different division of text into patterns
- providing two possible ways of dividing the original text into patterns as part of the initial data
- mixing the vocabulary from one text and the patterns of the other

A total of 504 trials were carried out (14 combinations, and 36 poems, 12 for each strophic form). Many of the resulting poems were either syntactically incorrect, or too short to be considered as poems. For this reason, evaluation took place in two stages.

During the first stage every resulting poem was assigned three numbers: (1) number of verses of the poem, (2) a value for its syntactical correctness, and (3) a value for its esthetical rating. These values are used as a first stage of filtering to avoid wasting evaluation effort on

verses that are too short or nonsensical verses unless they have a certain redeeming feature in an esthetical sense. Values were assigned on first inspection by the author.

Syntactical correctness was evaluated using the following scale:

1. the poem is mostly nonsense
2. the poem contains syntactic nonsense
3. the poem can be parsed as a weakly connected fragment of a sentence
4. the poem can be parsed as a strongly connected fragment of a sentence
5. the poem can be parsed as a connected whole

A fragment is considered weakly connected if it can be parsed in some way as a set of independent sentences. A fragment is considered strongly connected if at least some of the verses join together into sentences that make syntactic sense.

Esthetical rating was subjectively evaluated on the following scale <sup>1</sup>:

1. ugly
2. mediocre
3. acceptable
4. pleasing
5. very pretty

Table 3 shows the average results for each combination over the three different types of strophic form.

Poems rating lower than 3, 3, 3 on such a scale were not considered for the second stage of evaluation. This left a total of 45 poems to evaluate.

The second stage of evaluation was carried out by a team of volunteers. Evaluators were given a list of the 45 poems and they were asked to select the best five, and to assign them an order of preference. Each poem was assigned five points if rated first by some evaluator, four if rated second, three if rated third, two if rated fourth and one if rated fifth. The totals were added and the poems were ordered according to the resulting rating.

#### 4.4 Discussion of the Results

The results contain an enormous amount of information, only part of which has been mined at this stage. However, some very interesting conclusions can be drawn from the resulting facts.

Since it had been assumed from the start that if the choice of words and/or the choice of patterns play an important role in determining the quality of a poem, then

<sup>1</sup>Allowances were made for the fact that verses were the result of a computer program. The scale of 1 to 5 is taken as the bottom end of a 1 to 10 scale for human-generated poems.

Table 3: Validating Verse Draft

Combin.	Rating	Romance	Cuarteto	Terceto
1	Num	4	2	3
	Synt	3	4	3
	Aesthet	2	2	2
2	Num	3	2	2
	Synt	2	3	3
	Aesthet	2	3	2
3	Num	3	2	2
	Synt	3	3	3
	Aesthet	3	3	2
4	Num	1	1	1
	Synt	4	3	4
	Aesthet	3	2	2
5	Num	4	2	3
	Synt	3	2	3
	Aesthet	3	2	3
6	Num	2	2	3
	Synt	4	3	3
	Aesthet	4	3	2
7	Num	4	2	2
	Synt	3	3	3
	Aesthet	3	2	3
8	Num	5	2	3
	Synt	3	3	3
	Aesthet	2	2	2
9	Num	6	2	2
	Synt	2	3	3
	Aesthet	2	2	1
10	Num	3	2	2
	Synt	2	3	2
	Aesthet	1	2	2
11	Num	3	2	2
	Synt	3	3	3
	Aesthet	2	2	2
12	Num	0	0	0
	Synt	1	1	1
	Aesthet	1	1	1
13	Num	0	0	0
	Synt	1	0	0
	Aesthet	1	1	1
14	Num	0	0	0
	Synt	0	0	0
	Aesthet	1	1	1

there should be considerable differences between WASP poems obtained from one or the other set of data (poetic or academic). This hypothesis is validated by the fact that only six of the 45 acceptable poems were generated using the academic set of initial data.

Overall, only nine of the combinations that were tried managed to produce a poem that went into the final selection. Of these, only one of them was not using an ex-



tended version of the original vocabulary. However, that very one did produce the top scoring poem according to the evaluators. This suggests that in general terms the system performs better with a wider choice of vocabulary, unless the random factor in the generation process actually comes up with a poem that closely mirrors the original one (which is what happened in this case). While it is clear that recovering the original poem is bound to give an acceptable result, this is hardly a desirable solution.

Two of the combinations that produced most top scoring poems were working with vocabularies that had been extended with extra copies of words other than nouns, adjectives, adverbs or verbs (prepositions, articles, pronouns...). These words tend to appear more often than others in poems, and, being usually short, play an important role as cohesive element for longer words that are more difficult to fit into the metric.

Different divisions of the original text into patterns has no great influence on the result once the vocabulary has been extended. This is because an extended vocabulary contains words of different sizes for the same categories. With limited vocabularies, altering the length of verse patterns may result in the desired length not being achievable by combinations of the given words into the shorter patterns.

Mixing data from two different pieces of text (patterns of one, vocabulary of the other) can have drastic negative effects if there is no match between the categories that appear in the verse patterns and the categories represented in the vocabulary. However, it produces very interesting results from an aesthetic point of view. While no way has been found yet to evaluate this fact numerically, it has been observed informally by many of the evaluators and it should be taken into account for further analysis.

## 4.5 Further Work

The present experiment is intended as preliminary work in a long term project of developing a knowledge based poem generator in Spanish. The results obtained will help to discriminate between the different possible strategies. Additional knowledge and heuristics governing the selection of appropriate verse patterns to follow a given verse might be used either to guide poem construction or to eliminate poor results.

Several interesting insights have been obtained from the analysis of the results presented here. Such cases have been mentioned in the body of the paper wherever appropriate. Better heuristics must be developed for the selection of appropriate pattern for the next verse. Verse patterns should be distinguished in some way according to whether they are beginning, middle or end sections of a sentence. The evaluation procedures are still subject to a great deal of improvement. In a matter where subjective opinion of the reader, special effort must be made to devise an evaluation procedure that provides a rigorous

rating without interfering with the natural attitude of the evaluator as reader of a poem.

## Acknowledgements

The author wishes to thank to the magnificent team of volunteers that steadily worked their way through reams of verses (particularly obscure ones in many cases) to provide a reasonable evaluation of the performance of the system: Miguel Mulet Parada, Iñigo Eguzguiza, [Juan José Escribano Otero](#), Beatriz San Miguel López, Carlos Bezoz Daleske, Alberto Díaz Esteban, Luis Guerra Salas, Carlos Bruquetas, Oscar Rodríguez Polo, [María José García García](#), Javier García Navas, and Celia Rico Pérez.

## References

- P. Gervás, 'A Logic Programming Application for the Analysis of Spanish Verse', to be presented in: *First International Conference on Computational Logic, Logic Programming Implementations and Applications stream*, Imperial College, London, UK, 24th to 28th July, 2000
- T. Navarro Tomás (ed.), Garcilaso de la Vega. Obras, Espasa-Calpe, Madrid, 1973, Soneto XXIII, pp 225.
- A. Quilis, *Métrica española*, Ariel, Barcelona, 1985
- Real Academia Española (Comisión de Gramática), *Esbozo de una nueva gramática de la lengua española*, Espasa-Calpe, Madrid 1986.
- C. Rico Pérez, *Aproximación estadístico-algebraica al problema de la resolución de la anáfora en el discurso*, chapter 5, pp. 143, PhD Thesis, Departamento de Filología Inglesa, Universidad de Alicante, 1994.
- G. Riley, 'A Tool for Building Expert Systems', <http://www.ghgcorp.com/clips/CLIPS.html>
- V. G. Wiliamsen, J. T. Abraham, 'Association for Hispanic Classical Theater web page', <ftp://listserv.ccit.arizona.edu/pub/listserv/comedia/poetic1.html>