Automatic Generation of Poetry using a CBR Approach

Pablo Gervás
Dep. Sistemas Informáticos y Programación
Universidad Complutense de Madrid
email: pgervas@sip.ucm.es

Abstract: *A case based reasoning application that generates poetry versions of texts provided by the user is presented. Cases consist of a sentence of prose (used as retrieval key) associated with a corresponding poem fragment (used as starting point for the solution). Adaptation takes place by combining phonetic, metrical and lexical information about the words in the different sources - the prose message, the retrieved case, and additional vocabulary provided by the user. An extension of this system is proposed. Such extension would include semantic information in an integrated manner, allowing close interaction between the basic CBR processes - retrieval and adaptation - and the natural language semantics of the words involved.*

Keywords: Natural language generation, case-based reasoning

1 Introduction

The composition of poetry ranks among the most challenging problems of language generation, and is therefore a good test-bed for techniques designed to improve the quality of generated texts. Poetry has the advantage of not requiring exaggerate precision. To a certain extent, imposing restrictions over the form of a poem implies a slight relaxation on the specification of the content.

In order to model the process of poem composition two issues would have to be tackled: generation or selection of a content for the poem – the message to be conveyed –, and the production of a specific aesthetically pleasing form for that message – the final poem. These tasks may in real life take place simultaneously and there is surely a certain interdependence between them, however, it is only the second part that is really computationally tractable at present. The production of an aesthetically pleasing form involves: selection of a specific strophic form or stanza, which determines length of line in syllables, number of lines and distribution of the text over them, and a specific rhyming

pattern; finding appropriate rhymes, which impose restrictions on the words that can be used at the end of a line; and ensuring each line has the appropriate length and fits into a pattern of stressed and unstressed syllables (usually defined in terms of feet).

This general process will require in many cases either substitution of specific words for similar or related ones, or even full paraphrasing of significant portions of the initial formulation into constructions more suited to the specific strophic form sought. There is no easy algorithm for carrying out this task. The knowledge that a human writer puts into play when carrying it out might be formalised in terms of rules that code the required transformation. However, it is unlikely that professionals think of the process in these terms, and it would be a monumental endeavour to develop such a set of rules. Alternative approaches must be sought to simulate the process.

One established technique that allows the simulation of existing problem solving skills without resorting to coding the knowledge in terms of rules is case based reasoning (CBR), [AP94]. For the present application, a case would correspond to a given sentence in the source text format - the statement of the problem or the content for the poem - paired with the corresponding transcription onto the object text format - the solution of the problem or the resulting poem. Such a formulation stores the necessary information about how a particular sentence is broken up into lines or fragments of lines, how and where words are added or eliminated, and how the relative order of words is altered. Therefore, in order for the information captured in a case to be complete, one must explicitly store information about which words in the object version have been used in place of words that appeared in the source version - and which may have been dropped.

2 The automatic generation of poetry

Two different approaches have been attempted in dealing with automatic generation of poetry.

The work of Manurung [MRT01, MRT00] draws on rich linguistic information (semantics, grammar) to generate a metrically constrained grammar-driven formulation of a given semantic content. The generation of poetry is attempted as the sequence of an initial transcription of the corresponding message into a semantic representation of its content, followed by the generation of a poem corresponding to that semantic representation. Given the intuition that there is strong interaction between content and form during poetic composition by real people, this approach must surely lead to good modelling of the creative process. It has the disadvantage of being a knowledge-intensive approach to the problem, requiring strong formalisms for phonetics, grammar, and semantics, together with some form of modelling a certain aesthetic sense overlapping all three. In the present paper, an heuristic alternative is pursued.

The work of Gervás [Ger00a, Ger00b, Ger01b, Ger01a] draws on prior poems and a selection of vocabulary provided by the user to generate a metrically driven re-combination of the given vocabulary according to the line patterns extracted from prior poems. Several forward reasoning rule-based systems have been developed using different heuristic

approaches.

WASP [Ger00b], used a set of construction heuristics obtained from formal metric constraints to produce a poem from a set of words and a set of line patterns provided by the user. The system followed a generate and test method by randomly producing word sequences that met the formal requirements. Output was impeccable from the point of view of formal metrics, but clumsy from a linguistic point of view, and it made little sense.

The ASPID system [Ger00a] provided specific algorithms for the selection of a working set of words from an initial vocabulary using methods based on similarity calculations between the message proposed by the user for his poem and a corpus of already validated verses. Based on the similarity calculations, the system established a set of priorities over the complete available vocabulary. The next word to be added to the poem draft was initially looked for only among words marked with the highest priority, with the search extending in subsequent steps to words of lower priority only if none had been found in the previous step. This procedure improved search times considerably and it made possible computations with wider vocabulary coverage and narrower constraints on strophic forms. However, above a certain threshold (of vocabulary size and/or number of constraints imposed on the poem) even the method of establishing a priority ordering on the available words failed to ensure successful termination.

Both WASP and ASPID constructed poems by taking lines in prior poems as the basic building units for adaptation. This procedure resulted in a very agile construction mechanism, tailored to ensure strict metrical correctness and focusing closely on rhyme, but led to poor results from a syntactic and semantic point of view. The work presented here attempts to improve the quality of the results from the point of view of content by considering a sentence rather than a line as basic unit for adaptation.

3 A CBR Poetry Generation System

The automatic generation of poetry starts from the following data, which constitute an initial statement of the problem:

- a sentence (or set of sentences) in prose form to be converted into poetry,
- a specific stanza that the final poem must comply with,
- a set of cases for that specific stanza, and
- a set of words that may be used in the final poem.

ASPERA [Ger01a, Ger01b] is a forward reasoning rule-based system that is a working prototype for the described processes. ASPERA performs the following sequence of operations:

1. from a corpus of verse examples (cases) a specific case is retrieved (CBR Retrieve step) for each sentence of the intended message (the structure of the corresponding

case determines the distribution of the intended message over the chosen strophic form);

2. generates each of the lines of the poem draft by mirroring the POS structure of each of the lines of the chosen case - optionally combining in words from an additional vocabulary (CBR Reuse step) applying additional restrictions to enforce metric criteria;
3. presents the draft to be validated or corrected by the user (CBR Revise step); and
4. carries out an analysis of any validated poems in order to add the corresponding information to its data files, to be used in subsequent computations (CBR Retain step).

ASPERA is written in NASA's CLIPS rule-based system shell. The operation of the prototype is described below in terms of an example.

3.1 Dealing with text

Information taken into account about words is restricted to:

- their actual syntactic form and a part of speech tag associated with them - which encodes their grammatical role including number, gender and person
- metric data for each word - number of syllables, position of stressed syllable, rhyme, and restrictions on possible metric re-combinations

Therefore, every element of text is considered only at two levels: as a list of specific words, and as a list of the associated part of speech tags.

The part of speech (POS) tags are actually strings that represent information about part of speech, number, gender, and person of the words. This approach provides two separate views of a text. A text may be seen either as a list - the order of appearance is significant - of words, or as the corresponding list of POS tags. Of these two views, the first one is more specific and the second one more general. Two texts may have the same list of POS tags and yet have completely different meanings depending on the specific words chosen for the list of words that match these POS tags. Any system operating at this level of precision requires additional knowledge in the shape of a lexicon (associated pairs of word / part of speech tag) and a set of cases encoded at the appropriate level of precision.

As an example of how text is handled by the system, consider the simple text provided by the user as basic entry. The user submits a text which acts as a proposal of the message of the intended poem, for instance:

```
bebed los vasos de vino antes de que el camarero cierre
```

The proposal is analyzed in terms of part of speech (POS) tags:

```
bebed/VLPM2P los/ARTDMP vasos/NCMP de/DET vino/NCMS antes/ADVT
de/DET que/CQUE el/ARTDMS camarero/NCMS cierre/VLPM2P
```

3.2 Retrieving the correct case: poem structure

To avoid excessive proliferation of cases, some form of abstraction is required, to ensure that a case represents a number of sentences and not just the particular one from which it arises. This is achieved by letting every word stand in a case as a token for words of equivalent syntactic category.

A case will therefore consist of: an already built sample of the object text - the solution of the case - associated with the corresponding sample of source text - the description of the case -, where both description and solution of each case are only analysed in terms of words/parts of speech involved; together with a correspondence between the description and the solution.

The set of cases must include some that are reasonably similar to the problem statement (coverage of the case-base for the given problem space).

To achieve the level of abstraction described, the POS list for this proposal is used to retrieve the case whose prose description has a POS list is most similar to it. In the current version of the prototype, similarity is defined in terms of co-occurrence of the same POS tags and relative positions of related POS tags in the POS lists of the texts to be compared. The POS lists of the user's proposal and the POS list of the best matching case for the example given are presented below to show the type of similarity that is interesting for the system:

```
VLPM2P     ARTDMP NCMP DET       (NCMMS) ADVT DET CQUE ARTDMS NCMS VLPM2P

VLPM2P DET ARTDMP NCMP DET ARTDFS NCMFS  ADVT DET CQUE ARTDMS NCMS VLPM2P
```

Structural similarity has been preferred for simplicity. An important improvement of the system that is envisaged in the near future is to include semantic information in the form of an ontology for the vocabulary. This would allow a greater range of similarity measures and should noticeably improve system performance.

The basic information for the case whose similarity with the user input has been shown above is presented below:

```
(case (n acmamq) (beg nil) (end nil)
    (wordsProse
       disfrutad de los placeres de la juventud
       antes de que el tiempo pase)
    (POStagsProse
       VLPM2P DET ARTDMP NCMP DET ARTDFS NCMFS
       ADVT DET CQUE ARTDMS NCMS VLPM2P)
    (wordsPoetry
       *line*
            coged de vuestra alegre primavera
       *line*
            el dulce fruto antes que el tiempo airado
       *line*
            cubra de nieve la hermosa cumbre
```

```
   *line*)
(POStagsPoetry
   *line*
        VLPM2P DET ADJPOSFS ADJGFS NCFS
   *line*
        ARTDMS ADJGFS NCMS ADVT CQUE ARTDMS NCMS ADJGMS
   *line*
        VLPS3S DET NCFS ARTDFS ADJGFS NCFS
   *line*))
```

In order to operate correctly, the system requires information about how to break up a sentence over a number of lines. This is achieved by including this extra information in the definition of a case. A case is made up of a prose part (which acts as key for retrieval when searching for a case that matches the proposed message) and a poetry part (which is recorded in poem form, line breaks being significant, and which is used in the construction of the solution).

The case provides a structure for the poem to be generated as a solution (as given in the `POStagsPoetry` part of the case). The structure of the poem is currently decided exclusively during retrieval. Whereas a human poet may be creative in the structuring of his poem, the prototype is at present restricted to retrieving structures already used before. Once a language ontology has been incorporated, the additional information it supplies may provide the means for limited adaptation of the retrieved structures.

3.3 Suggesting words to fill the structure: vocabulary

The structure provided by the retrieved case must be filled during adaptation with new words that are appropriate to the message proposed by the user. The message itself will usually not provide enough words - or the right kind of words - for a poem to be built. In the current prototype, the user has the additional responsibility of providing a task-specific vocabulary selection tailored to ensure that the set of words available for adaptation is: 1)wide enough for adaptation to be possible, and 2) specific enough for results to fall within acceptable bounds of discrepancy with respect to the problem statement.

A significant contribution to the creativity of the system depends on the selection of these words.

System use and evaluation of results has shown that a good set of additional words for the example given is:

```
(vocabulary
    tomad      embriagadora    ba_quica
    to_nica     bebida          rojo
    ganimedes   mesonero        hostelero
    posadero    hombre          feo
    ciegue      ira             divina      fuente
)
```

The arguments behind this specific choice are as follows. Some words in the proposal are similar to words in the case description, but find no place in the structure of the case solution. These additional words are related in different ways to the ommitted words, but have the right characteristics to fit into the chosen structure.

Our present efforts are directed towards formalizing this type of argument. In achieving this, much depends on identifying the meanings of 'being related to' that are useful in each case. Again, the use of an ontology may provide the key to this question. The cases can play an important role in this process. If relations of the kind embodied in the ontology can be traced between the prose in the case description and the poetry in the case solution, then words similarly related to the user's proposal may be used to fill the poem structure. The case would then be providing not only the structure but also the key to finding the right words through the ontology.

3.4 Adaptation

In general terms, adaptation takes place by replacing as many as possible of the words in the case solution with words from the intended message, and filling in any gaps – if the need to satisfy the metric requirements precludes the use of words either in the case solution or in the intended message – with words in the additional vocabulary. The main guiding principle in this process is the relative position of words with the same POS tag in the intended message and the case solution, together with additional criteria described below. The choice of similarity criteria employed during retrieval is designed to maximise the number of words with the same POS tags in similar relative positions.

The basic generation algorithm is applied to each line of the retrieved structure, which acts as a *skeleton solution* for the corresponding line of the poem, and the available words. The elementary generation units are the skeleton solution being followed (which acts as guiding form) and the *current draft of the solution*. At each step of the generation the words are chosen to match the next POS tag in the skeleton solution.

Certain heuristic guidance is required during adaptation to avoid the pitfalls associated with the lexical approximation - namely, the high risk of obtaining meaningless results. To provide it, cases are annotated with information regarding which elements of the source text are left out of the object text component, which elements are used (and in which specific location), and which elements of the object text are not related to the source text and must be provided for the solution from an alternative source. This information is represented in terms of the appearance or appearances of the candidate word in one or more of the following: the problem statement, the case description, and the case solution. With respect to this information, words available to the system can be classed in to seven categories:

A. in problem statement, in description, in solution
B. in problem statement, not in description, in solution
C. in problem statement, in description, not in solution
D. not in the problem statement, in description, in solution
E. not in the problem statement, not in description, in solution

F. not in the problem statement, in description, not in solution

G. not in the problem statement, not in description, not in solution

Of these, the first level represents an ideal substitution. The second level captures additional possibilities to ensure that any words not matched by the description but with a possible role in the solution are not lost in the adaptation. These first two levels are not usually enough to provide candidates for all positions in a possible solution, and the remaining levels drive the assignment of extra words. Words in category G are drawn from the additional selected vocabulary provided for the specific problem. Words corresponding to categories A,B, D and E have a specific place in the solution for which they are well suited. The words in the other categories, if they are included, must be guided by the adaptation process to a relative position in the solution as close as possible to the corresponding relative positions in the sources. In addition to these basic categories, there are also words that the system in its present form is unable to include in the solution, even if they appear in both the problem statement and the case description. This happens in instances where there is no POS tag in the solution matching the desired word. The adaptation process is restricted to filling in the skeleton solution provided by the case, this solution being the ultimate judge of what is valid in a final result.

Different levels of priorities can be assigned to these words categories in order to guide the word-for-POS-tags substitutions during adaptation. The assignment of priorities constitutes the driving heuristic for determining the final result. For instance, considering words for substitution in the following order (A, B, C, D, E, F, G) is equivalent to aiming for a solution as close to the retrieved case as possible. This assignment gives rise to texts which are very similar to the ones in the case-base, and is therefore a good conservative option, yielding not very original results which follow very closely the required format. An alternative assignment, based on the idea of obtaining a solution as different as possible from the retrieved case, would use up the word categories in the following order (A, B, C, G, F, E, D). This alternative can result in a relatively creative output, and therefore runs a higher risk of producing texts which are somewhat lacking in sense in spite of all the heuristics applied.

Additionally, the candidate word must fulfill a number of additional constraints related to the poetic form chosen by the user.

The constraint on metric pattern of each line imposes additional restrictions on the construction process used during adaptation. On one hand, the basic unit for the generation algorithm must shift from the sentence to the line, because the metric patterns to be obeyed are defined over lines and not over sentences. On the other hand, simple substitution based on priorities established beforehand is no longer enough, because metric criteria have to be taken into account as well when deciding whether a word is an acceptable candidate for substitution. The correct metric considerations for the desired stanza must be imposed as additional constraints on the substitution process during adaptation.

The constraint on the rhyme of words at the end of a line imposes a double restriction: on one hand, final words of each line must be chosen with this additional constraint in mind, and on the other hand enough words with the required rhyme must be available in the vocabulary to fit all the rhyming positions in the desired strophic form. It is usually

an important challenge to meet this requirement as well as the restrictions on the content of the poem.

For the system to be able to handle these restrictions, the lexicon must be extended to include metric information for each available word - number of syllables, rhyme, and whether it is open-ended or not with respect to syllabic recombination with adjacent words.

If the chosen word meets the constructive requirements (position of stressed syllables and appropriate rhymes), it is appended to the draft of the solution and the process is iterated.

The current version selects the required word at each stage solely based on its syntactic category and its relative position in either user proposal, case description or case solution. This works reasonably well for words originating from these sources, but not so well for the additional vocabulary. It is hoped that having access to the sort of relations embodied in an ontology – mentioned above in the context of selecting the additional task-specific words that must currently be provided by the user – also provide the means for finding the most suitable position in which to place them during adaptation. Additionally, as mentioned above, the ontology may provide the information needed to modify the structure selectively, for instance by replacing a masculine singular noun with a femenine singular noun if they refer to the same concept.

Additionally, an ontology would play an important role in dealing with the important problem of *ambiguity* in the meaning of words. Having an adequate conceptual representation of the words being used would provide the means to *disambiguate* between different meanings of the same words during these processes.

An example of the sort of poem that the system might build for the input data described would be:

```
(sample_output
    tomad de vuestra ba_quica bebida
    el rojo vino antes que el hombre feo
    ciegue de ira la divina fuente )
```

4 Conclusion

The transformation of a given prose text into a poem can be represented as a CBR process [Ger01a, Ger01b], where each case contains a sample – a sentence – of the source text as case description and a sample of the object poem as case solution. This paper presents an enriched version of this method, taking into account the possibility of assigning different priorities to the different possible sources for words but still restricted to applying only lexical information about the words involved, has been implemented in a working prototype. Following this approach, creativity regarding poem structure and word selection is discussed in terms of the retrieval and adaptation processes taking place.

In view of the various arguments outlined so far, it is perceived that the system would greatly benefit from incorporating semantic information in the form of a knowledge rich

ontology. This addition must be done in some way that allows the new information to play a role easily in the various processes carried out by the system.

Specific operations based on this additional information could play very useful roles, when applied to the available vocabulary with respect to the words either in the description of the requested content or in the cases, if they can made to interact with the various processes involved in the application. During retrieval a semantic description for words introduces the possibility of recovering cases where a similar meaning is conveyed with completely different syntactic constructions. Additionally, it could act as a mechanism for extracting from particular cases the relevant relations between case description and case solution, to be used in selecting an adequate vocabulary. During adaptation these same relations can be used to drive the use of words from the additional vocabulary which are similar to the words in the input.

The foreseeable advantages of enriching the approach with a knowledge rich ontology for the words being used have been described at each stage, and the future work that is envisaged along these lines has been outlined.

References

[AP94] Agnar Aamodt and Enric Plaza. Case-based reasoning: Foundational issues, methodological variations, and system approaches. *AI Communications*, 7(i), 1994.

[Ger00a] Pablo Gervás. Un modelo computacional para la generación automática de poesía formal en castellano. *Procesamiento de lenguaje natural*, 26(26), 2000.

[Ger00b] Pablo Gervás. Wasp: Evaluation of different strategies for the automatic generation of spanish verse. In *Proceedings of the AISB-00 Symposium on Creative & Cultural Aspects of AI*, pages 93–100, 2000.

[Ger01a] Pablo Gervás. Creativity versus faithfulness. In *Proc. of the AISB-01 Symposium on AI and Creativity in Arts and Science*, 2001.

[Ger01b] Pablo Gervás. An expert system for the composition of formal Spanish poetry. *Journal of Knowledge-Based Systems*, 14(3–4):181–188, 2001.

[MRT00] H. M. Manurung, G. Ritchie, and H. Thompson. A flexible integrated architecture for generating poetic texts. Informatics Research Report EDI- INF-RR-0016, University of of Edinburgh, 2000.

[MRT01] H. M. Manurung, G. Ritchie, and H. Thompson. Towards a computational model of poetry generation. In *Proc. of the AISB-00 Symposium on Creative and Cultural Aspects of AI*, 2001.