Generating Poetry from a Prose Text: Creativity versus Faithfulness

Pablo Gervás
Departamento de Sistemas Informáticos y Programación, Universidad Complutense de Madrid
Avenida Complutense s/n
28040 MADRID, SPAIN
pgervas@sip.ucm.es

Abstract

ASPERA is a case based reasoning application that generates poetry versions of texts provided by the user. The cases on its case base consist of a sentence of prose (used as retrieval key) associated with a corresponding poem fragment (used as starting point for the solution). Each case includes information about how the words in the prose text are related to the words in the poetry solution. The system applies a generation process guided by metrical rules to adapt its best matching cases to the sentences in the text provided by the user. ASPERA is a refinement of an earlier system and its casebase is currently being extended. It is hoped that soon it will have enough coverage for the system to become fully operational. The issue of whether such as system should be creative (surprising the user with poems not explicitly determined by his proposal) or faithful (keeping as close as possible to the proposal given) is discussed with a view to including in the system a facility to control its behaviour in this respect.

1 Introduction

There are certain situations in which a given message, already formulated as a prose text, needs to be reformulated in order to convey it in some poetic form. This situation may arise as a basic step in the process of composition of specific forms of poetry (poems in praise of someone or inspired by a certain event), but it is specially apparent in the case of translations of a poem in one language into a supposedly equivalent poem in a different language. What the run of the mill translator obtains from the original poem is, at best, an exquisite rendition of the message of the original poem in a new language. The particular traits that conformed the original poem get lost in the translation. If the translator wants to recreate the effect of the original, this new version in the new language must be modelled into the corresponding poetic form in that language.

Although this is not a good approximation for a cognitive model of human poetry writing, the nature of the application brings to the fore an important issue that tends to be obscured when tackling computer creativity from other points of view: should one favour approaches that come up with new material (recognisably different from the initial text) or should one aim for a final product as close as possible to the original? There is a great tendency to reduce the idea of creativity to 'producing something that was not already available'. In the realm of poetry (and language in general), such a definition is too restricted. Random generation aided by a basic grammar will easily produce 'poems' that nobody has ever seen before, but seriously lacking in content. It is clear that something else beyond innovation is required. In order to model completely the human creative process of poem composition two issues would have to be tackled: generation or selection of a message (or concept) for the poem, and the production of a specific aesthetically pleasing form for that message. 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.

This process requires:

- selection of a strophic form
- distribution of the message across the chosen strophic form
- shaping the lines to an adequate length and metric
- finding appropriate rhymes

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. Many options are open, and the secret for achieving a version acceptable as a poem lies in making the correct decisions at this stage. If one starts from a very clear concept of what is to be said or how it should be said, the aim during composition is to obtain a poem that reproduces in as much as possible the style of the original formulation. Creativity has then been delegated to the conceptual design of the poem. However, if composition were to be fully creative, it ought to allow some form of controlled departure from the original formulation.

2 Previous Work

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

The work of Manurung (2000a; 2000b) draws on rich linguistic information (semantics, grammar) to generate a metrically constrained grammar-driven formulation of a given semantic content.

The transcription of a text in prose form into its poetic equivalent can be 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.

2.1 Heuristic approaches to poetry generation

The work of Gervas (2000a) 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.

WASP (Gervas, 2000a) was a forward reasoning rule-based system that used a set of construction heuristics obtained from these 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 results were impeccable from the point of view of formal metrics, but they were clumsy from a linguistic point of view and made little sense.

The ASPID system (Gervas, 2000b) 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.

ASPERA (Gervas, 2000c) is a forward reasoning rule-based system that obtains from the user basic style parameters and an intended message; applies a knowledge-based pre-processor to select the most appropriate metric structure for the user's wishes; and, by intelligent adaptation of selected examples from a corpus of verses, carries out a prose-to-poetry translation of the given message. In the composition process, ASPERA combines natural language generation and CBR techniques to apply a set of construction heuristics obtained from formal literature on Spanish poetry. If the user validates the poem draft presented by the system, the resulting verses are analysed and incorporated into the system data files.

2.2 Modelling literary style

Close analysis of the subject of literary style is far beyond the scope of state of the art language manipulation, however a simple approximation may be obtained by stripping the representation of style down to the frequency groupings of types of words - e.g. particular syntactic categories - and the appearance of particular choices of words. This is indeed done for statistical analysis of texts in cases of disputed authorship (Stratil and Oakley, 1987; Tankard, 1986), in an attempt to identify the - unknown - author of a specific text by comparing the results of its analysis with the corresponding results for texts of known to have been written by specific authors.

3. The Prose to Poetry Transcription

There is doubtless a creative process involved in transforming a prose text into poetry. This process can be described in terms of translating from the 'language' of prose to the 'language' of poetry. This translation is peculiar in that some very important features from the linguistic point of view must remain the same: not only the content of the text but also the actual language - for instance English, Italian or Spanish -, and many of the words. The parameters that change most are rhythm, rhyme, metric, distribution over lines, word order... Like all processes of translation, there is no easy algorithm for carrying out the task. If one considers the process in some detail, this metaphor can be refined even further. Each particular stanza imposes specific constraints on rhythm, rhyme, distribution over lines, and metric. Therefore, one could define the required form for that stanza as a specific 'language' into which prose can be translated.

The knowledge that a poet puts into play when carrying

dated - or modified - by the user (Revise step) and incorporated into the case base to be available for further problem solving (Retain step).

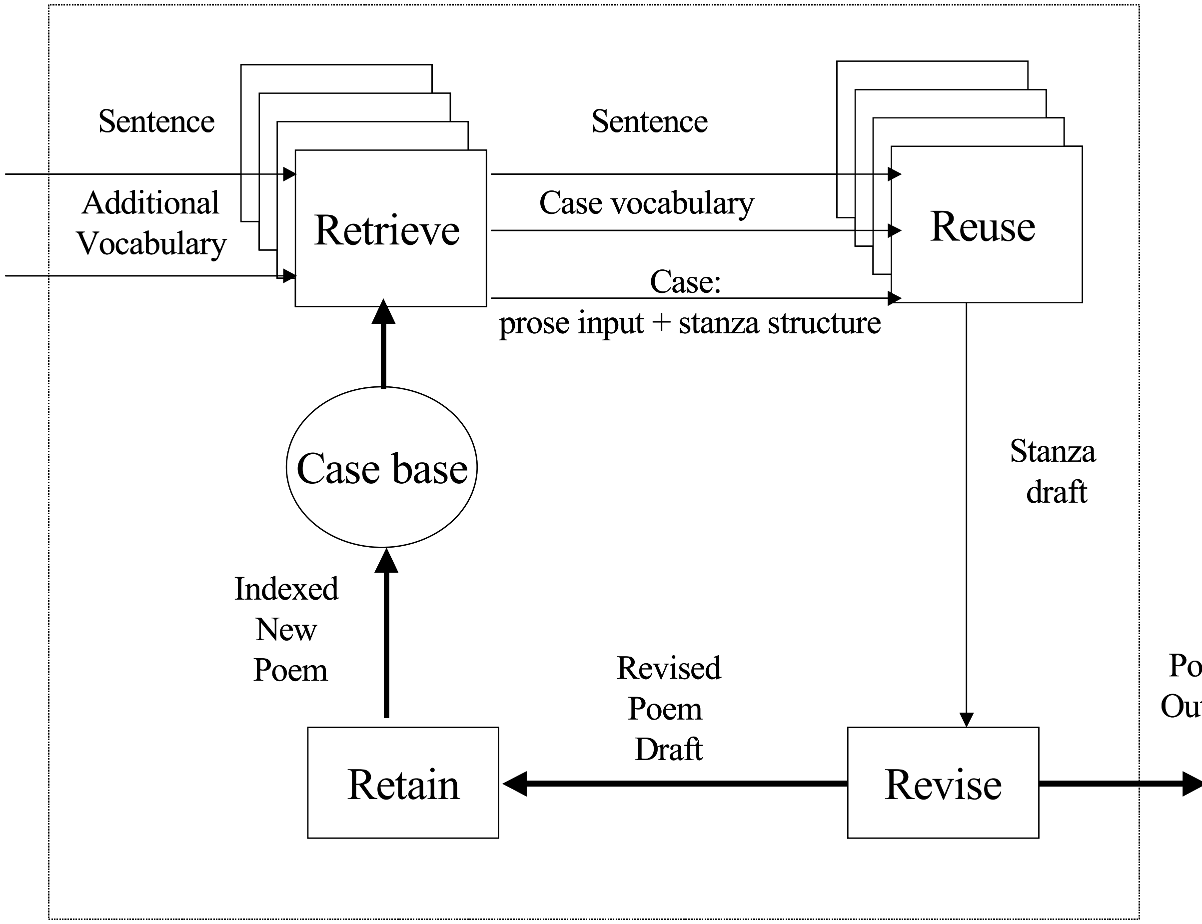

Figure 1 Verse generation using CBR

out this translation might be formalised in terms of rules that code the required transformation. However, not even accomplished poets 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). Case based reasoning (Aamodt and Plaza, 1994) resorts to storing a set of previously solved problems, each coupled with the corresponding solution. These are known as cases. Faced with a new problem, a case is retrieved that best matches it (Retrieve step). The corresponding solution is adapted to the new problem (Reuse step). In order to achieve incremental development of the case base, the resulting solution must be vali-

For the present application, a case would correspond to a given sentence in prose form - the statement of the problem - paired with the corresponding verse transcription - the solution. A sentence is chosen as the basic unit in order to ensure that a certain semantic and syntactic correctness prevails over the whole poem. Previous versions of the system (see Gervas, 2000a; 2000b; 2000c) constructed poems by taking lines in prior poems as basic units. 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 move to considering a sentence as basic unit is an attempt to improve the quality of the results from the point of view of content. Poem lines still remain an important building block to be taken into account. The present system requires additional information about how a sentence is

broken up into lines. This information must be included in the representation of each case.
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 as a token for words of equivalent syntactic category. 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. In order for this information to be complete, a case must also store information about which words in the poetry version have been used in place of words that appeared in the prose version - and which may have been dropped. The generation of such cases from a given sample problem constitutes the retain step of the CBR process.

4. A CBR Poetry Generation System

ASPERA is a forward reasoning rule-based system that performs the following sequence of operations:

1. from a corpus of verse examples (cases) a specific case is retrieved files (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);
3. presents the draft to be validated or corrected by the user (CBR Revise step); and
4. carries out a linguistic 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).

The overall structure of the ASPERA system is presented in Figure 1.
ASPERA is written in CLIPS, a rule-based system shell developed by NASA.

4.1 Building a case base

Two issues become important when a casebase built up of such cases is to be used for simulating the process of prose to poetry transcription: how to retrieve the appropriate cases for a given initial prose text, and, once found, how to reuse their solutions to produce the poetic transcription.
The problem starts from the following data, which constitute an initial statement of the problem: a set of sentences in prose form that need 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, grouped into equivalence classes. The concept of equivalence class used here is an informal one that covers any type of relationship between words that allows one to be substituted by another without significant modification of the meaning of the message conveyed. The initial data must fulfil the following constraints:

- the set of available words - the vocabulary - must include the words in the statement of the problem and the words in all the cases, as well as additional associated words that may be used to fill in gaps as required
- the set of cases must include some that are reasonably similar to the problem statement (coverage of the casebase for the given problem space)

4.2 Criteria for retrieval of previous cases

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 used in the construction of the solution).
The retrieval of relevant cases is carried out in two stages. The first stage filters an initial set of candidate cases for each sentence in the problem statement based on similarity calculations between the sentence itself and the statement of the problem for each case. The second stage ensures that the solutions of the cases found for each successive sentence in the message match up to give rise to a correct and complete stanza of the required type. This corresponds in fact to the planning step involved in a typical natural language generation process - the intended message is distributed over the actual structure of the linguistic contribution that is to be produced.

4.3 Adaptation of the selected cases

Once an appropriate set of cases has been found, the corresponding poetry parts are all linked up into a skeleton of the solution. To abstract away from each particular case being used, only the linguistic categories of the words in the case are taken to make up this skeleton solution, which then acts as a seed for the new poem.
Certain guidance is required during adaptation. To provide it, cases are annotated with information regarding which elements of the prose component are let out of the poetry component, which elements are used (and in which specific location), and which elements of the poetry solution are not related to the prose proposal and must be provided for the solution. This information is used by chaining these data with equivalent relationships extracted automatically by the system from the pair constituted by the prose proposal and the prose key of the case. Once the correct links have been established between

items appearing in the proposal and items that have to appear in the solution, adaptation can be carried out.

Adaptation takes place by successively substituting the tokens that stand for each type of word in this skeleton for words in the available vocabulary. The correct metric considerations for the desired stanza are imposed as additional constraints on this substitution process.

Substitutions are attempted in the following order or priority:

- words standing in a similar position in the original statement of the problem
- words standing in different positions but marked in the corresponding case as having been moved during transcription
- words in the same equivalence class as any of the above
- words in the associated vocabulary provided to fill in any gaps
- words in the solution of one of the cases being used

For reasonably faithful results, most substitutions should take place under the first three of the categories. If creative composition is desired, the last two alternatives allow the system to produce solutions that are not necessarily very close to the proposed message. In this way, they also act as default solutions for non terminating situations arising from poor coverage of the casebase for the given problem space.

At its present stage the system has no means of controlling explicitly the amount of freedom that is allowed in the construction process.

4.4 User intervention

The user cannot influence the composition of a poem while it is in process. However, the procedure described so far need not be understood as a fully deterministic process that substitutes the human intervention in the creative process. As mentioned above, the procedure relies heavily on a wise selection of the vocabulary and the casebase on which it operates. User control of the composition is restricted to the selection of input parameters. This need not discourage those who dislike the idea of a black box poem generator blind to the user's reactions. Because the generation process takes very little time, the user is encouraged to experiment with different selections of vocabulary or cases in the casebase, manipulating the initial data in search of satisfactory results.

Specific data that the user may modify in successive iterations include the particular wording of the problem statement, the definition of equivalence classes for particular words - adding or eliminating different ways of saying the same thing -, the selection of vocabulary and the selection of cases.

The system takes an integer number as input which determines how far from its original position in the initial formulation (relative to the overall text) it move to its final destination in the poem draft. The number provided is taken as a starting value and it should be very low if faithfulness to the message provided is the main aim. In cases where no fitting word is found, the system can progressively increase this value in search for a match. To avoid results that differ widely from the intended message, a maximum value is also provided. Both values allow control over the amount of freedom the system is applying in each case.

At a different level of interaction, the user can modify the solutions proposed by the system during the Revise step of the CBR cycle. In this way, undesirable features of the proposed solution may be eliminated from the new case that is updated into the casebase. Successive iterations can resort to the new case instead of the one used in the previous iteration, and may in this manner come up with more satisfactory results.

5. Putting the method to the test

The ASPERA system (Gervas 2000c) was originally developed for a casebase of classic Sixteenth century Spanish formal poems, using poem lines as basic construction unit. The architecture presented in this paper constitutes a major improvement in as much as it enriches the casebase on which the system relies by including not simply already made and validated poems, but complete cases, with a prose key, a poetry proposal for the solution, and specific information for adaptation purposes. As an additional advantage, the basic construction unit has been changed from a poem line to a prose sentence, which avoids a clumsy prose-poetry interfacing problem with earlier versions (the set of words in the intended message had to be distributed over the possible poem lines, requiring either complex heuristics or high computation time to achieve reasonable results).

As a distinct disadvantage, development of the proposed casebase for the revised system requires much more refined work, and, although well under way, it has not reached yet the threshold size necessary to provide coverage for significant experiments to be carried out.

The method proposed is to be tested over a casebase generated from existing texts that have both a prose and a poetry version. Translations of a poetic work in one language (Italian) into equivalent versions in another (Spanish) are used as sources. Each case is built by associating a given sentence from the prose translation with corresponding set of lines from a verse translation.

5.1 Construction of a set of test data

Four different translated versions (Chiclana, 1999; Crespo, 1977; Echeverría, 2000; Petrocchi and Martínez de Merlo, 1999) of Dante's Divine Comedy are considered. The original work is written in Italian verse. One of the translated versions renders it as straight prose. The other three present poetry versions of varying resemblance to the original strophic form. Three different casebases are generated based on equivalent fragments of the text. All of them include the corresponding fragment of the prose version as statement of the problem. In each one the specific verse rendition is included as the solution. Each individual casebase encodes a particular approach to the transcription of prose into poetry for a particular set of sentences. They have the advantage of all being based on the same problem statement - the prose version - and all of them complying to the same stanza. This qualifies them as a very good test benches for the proposed simulation of the transcription process.

5.2 Creativity versus Faithfulness

Several parameters influence the amount of freedom that the system has when composing a poem draft:

- whether strict metric rules are imposed on the poem (specific rhythm)
- whether the system is allowed to resort to additional vocabulary outside that presented by the user with this intended message

Additionally, the amount of freedom that the system must take (whether desired or not) is also influenced by:

- coverage of the case base with respect to the proposed message
- coverage of the additional vocabulary with respect to the structures present in the selected cases

The system may at a later stage be extended to include some means of controlling explicitly the amount of faithfulness or creativity that the user requires.

5.3 Turing Test Evaluation

The method is tested by attempting to generate a verse rendition of the original fragment using each one of the casebases in turn, and comparing the result with the verse rendition from which the casebase was extracted.

A rough version of the Turing test is carried out, with human evaluators being asked to compare automatically generated versions of a given prose fragment with the equivalent verse fragments produced by human translators.

Evaluation is carried out by generating a set of poetry versions and presenting the results to a group of evaluators. Each evaluator is presented with a prose fragment and a set of poetic transcriptions. The poetic transcriptions may include both the corresponding verse fragments of some of the original verse editions and verse transcriptions generated automatically. Evaluators are asked to rate verse renditions according to their faithfulness to the original prose and their poetic quality. They are also asked to identify the versions that have been automatically generated.

6. Conclusions

The simulation of the prose to poetry transcription presented here is not intended as a faithful model of the way human translators approach the task. Instead, it is a heuristic approximation to the task that makes the most of a set of previously solved examples, while allowing the user to control the input parameters that influence the final result. Even so, the simulation is conceptually based on a procedure universally employed when not-specially-talented individuals need to personalise a song, for instance, for a birthday, a wedding, or a particular event: pick a song that everybody knows and rewrite the lyrics to suit the situation under consideration. The procedure presented here employs this method at sentence level, and enriches it by providing - through an objective retrieval method with a built in stanza planning step - the means for combining sentences from different origins. This particular approach to the problem of generating customised lyrics or poetry is adapted to a formal CBR architecture, thereby providing a substantial methodological backbone. Additionally, the Revise and Retain steps of the CBR cycle provide the means for the system to progressively increment its casebase, thereby improving its coverage. By interacting iteratively with the user the system can learn different and better ways of solving the transcription problem than those that were originally available.

Acknowledgements

I would like to thank two anonymous referees for their helpful comments on an initial draft of this paper.

References

A. Aamodt and E. Plaza, Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches. *AI Communications*, 7(i) (1994), 39-59.

A. Crespo, Dante Alighieri, *Divina Comedia*, (Spanish verse translation), (Clasicos Universales), Planeta, Barcelona, 1977

A. Chiclana, Dante Alighieri, *Divina Comedia*, (Spanish verse translation), (Austral), Espasa Calpe, Madrid, 1999.

A. Echevarría, Dante Alighieri, *Divina Comedia*, (Spanish verse translation), (Literatura), Alianza Editorial, Madrid, 2000.

G. Petrocchi and L. Martínez de Merlo, Dante Alighieri, *Divina Comedia*, , (Spanish verse translation), (Letras Universales), Cátedra, Madrid.

P. Gervás, WASP: Evaluation of Different Strategies for the Automatic Generation of Spanish Verse, in: Time for AI and Society, Proceedings of the AISB-00 Symposium on Creative & Cultural Aspects and Applications of AI & Cognitive Science, 17th-18th April 2000, University of Birmingham, England, 93-100.

P. Gervás, 'Un modelo computacional para la generación automática de poesía formal en castellano', in: Actas del XVI Congreso de la SEPLN (Sociedad Española para el Procesamiento del Lenguaje Natural), Vigo, España, September 2000.

P. Gervás, An Expert System for the Composition of Formal Spanish Poetry, The Twentieth SGES International Conference on Knowledge Based Systems and Applied Artificial Intelligence, ES 200011-13th December 2000, Peterhouse College, Cambridge, England.

H. M. Manurung, G. Ritchie and H. Thompson, Towards a computational model of poetry generation, in: *Time for AI and Society*, Proceedings of the AISB-00 Symposium on Creative & Cultural Aspects and Applications of AI & Cognitive Science, 17th-18th April 2000, University of Birmingham, England.

H. M. Manurung, G. Ritchie and H. Thompson, A Flexible Integrated Architecture for Generating Poetic Texts, Informatics Research Report, EDI-INF-RR-0016, Division of Informatics, University of Edinburgh, May 2000.

M. Stratil and R.J. Oakley, A Disputed Authorship Study of Two Plays Attributed to Tirso de Molina, *Literary and Linguistic Computing*, Vol. 2, No. 3, 1987, pp 153-160.

J.Tankard, The Literary Detective, *Byte*, February 1986, pp 231-238.