Exploring Quantitative Evaluations of the Creativity of Automatic Poets

Pablo Gervás[1]

Abstract. The purpose of this paper is twofold: to show the practical applications of theoretical evaluation measures designed to capture the degree of creativity of a program, and to use the results to evaluate an effort to develop an automatic Spanish poet. Existing efforts for the development of automatic poets are described, and the implications of their particular architectures to the evaluation issues discussed are considered.

1 INTRODUCTION

The community of researchers devoted to the study of creativity has recently grown from a small group of people who worked on isolated projects to reach an important number of groups addressing issues in different domains and with a different focus. In this process, there is a need for some kind of quantitative means of evaluating the *amount* or the *quality* or the efficiency of a creative system.

The need to have objective measures is crucial in a general sense if we are to achieve the development of testable and comparable solutions to the problems that are being faced. In a field with as much subjective content as that of creativity, it becomes paramount not only to define some means of establishing quantitative measurements, but also to apply such measurements systematically to the designed solutions at each stage, in order to obtain from them guidance and stable references on which to base further development.

In recent times, research efforts in the field of creativity have produced a number of systems that attempt tasks that had so far been considered to be too creative for computers to tackle, such as musical composition, theory formation, or poetry writing. The resulting increase in interest from the research community has produced the very beginnings of a theoretical body of work on the evaluation of creativity. If the field is to progress steadily, the next step ahead is to apply these initial theoretical efforts to the practical systems being developed. The purpose of this endeavour should be two fold. On one hand, it should provide quantitative metrics on the creative behaviour of systems that may play a role in guiding subsequent design and development efforts. On the other hand, it should at the same time constitute a test of the suitability of the theoretical evaluation methods that have been proposed so far. While both theoretical and practical advances are valuable in a field as young as the study of creativity as it bears on computer programs, it is really in bringing together theory and practice that positive progress will be consolidated in the field. Practical systems should be tested according to the theories that are being put forward, and theoretical proposals should be applied to real cases to see if they adequately address the issues that are of import in the development of systems.

[1] Universidad Complutense de Madrid, 28040 Madrid, Spain, email: pgervas@sip.ucm.es

The present paper brings together a number of theoretical proposals put forward in the past few years and the practical creative systems developed in the past for the particular domain of poetry composition. Important issues concerning this particular domain are discussed, the different approaches are compared, and different proposals for the evaluation of creative behaviour are tried out against the results of one particular system.

2 EXISTING FORMALISATIONS OF CREATIVITY MEASUREMENT

If creativity and engineering are to collaborate successfully, some explicit way of measuring the activity involved in creativity must be established. Already Boden [2] distinguishes between H-creativity - the result is absolutely new in historical terms - and P-creativity - the result is new for the creator independently of whether it had been done before. In the face of this type of distinction, it is important to establish criteria that can be applied to a system and which take into account not only what is being created but also what was already available to the program when it started operating. Additionally, the measurements must take into account the need for a balance between creating artefacts that meet the general requirements within a given domain (are typical of the genre) and creating artefacts that are innovative (bring something new to the domain).

There are currently several proposals about how creativity might be measured quantitatively, at least indirectly if not directly, and in terms of certain qualifying functions for the specific domain in which the creative system to be evaluated is operating.

2.1 Assessing Creativity Based on How Good and How Typical the Results Are

Ritchie [13] provides an initial set of relevant concepts and 14 criteria based on those concepts for deciding whether a program is creative or not, which constitute a strong starting point for discussion. Two basic aspects are relevant: novelty (to what extent is the produced item dissimilar to existing examples of that genre) and quality (to what extent is the produced item a high-quality example of that genre). To measure these aspects, two functions are introduced: *typ*, which rates the typicality of a given item (item is typical), and *val*, which rates its quality (item is good). These functions take the form of *rating schemes*, which assign points in the interval [0,1] on a given property.

Another important issue that affects the assessment of creativity in creative programs is the concept of *inspiring set*, the set of (usually highly valued) artefacts that the programmer is guided by when designing a creative program.

According to this vision, the construction of a creative program follows a sequence of steps that can be instantiated for any particular

case: 1) select a set of basic items which are to guide the construction of the creative program (inspiring set) 2) map from the inspiring set to a program 3) establish: initial data values (possible parameters for the algorithm; a tuple of sets, each set being the range for one parameter to the generating procedure), generating procedure (given a tuple of initial data values produces a set of basic items).

Program construction follows therefore a basic scheme divided in two processes: one of selection (from basic items get inspiring set) and one of construction proper (define initial data ranges and procedure, initialise, run, and obtain a result set). The *initialisation* refers to selecting a choice of initial parameters, based on the inspiring set and the rating schemes. A *run* of the program is understood as the set of initial parameters together with the set of results.

Fourteen criteria are provided, relating the inspiring set and the set of results (and the corresponding subsets defined by applying to it the valuation functions defined above). A brief overall description of their intention is given in table 1.

Table 1. Description of basic criteria

Crit.	Description
1	All elements in the result are reasonably typical
2	A reasonable proportion of the results should be very typical
3	All element in the result are reasonably good
4	A reasonable proportion of the results should be very good
5	A reasonable proportion of the very typical results should be very good
6	A reasonable proportion of the results should be very good and not very typical
7	A reasonable proportion of the not very typical results should be very good
8	There should be a reasonable proportion between the very good and not very typical results and the very good and very typical results
9	A reasonable proportion of the inspiring set should appear in the results
10	A reasonable proportion of the results should not appear in the inspiring set
11	Results not in the inspiring set should be typical
12	Results not in the inspiring set should be very good
13	A reasonable proportion of the results is not in the inspiring set and very typical
14	A reasonable proportion of the results is not in the inspiring set and very good

Specific parameters are provided in the mathematical formulation of these criteria to control what is actually meant by "a reasonable proportion". The various criteria are intended as a box of tools from which to pick and chose a selection for a particular purpose.

2.2 Evaluating the Degree of Fine tuning

Colton et al [3] describe the effect on a program's creativity of the amount of knowledge that is taken as a starting point in whatever generation process it carries out. This is done based on the work of Ritchie [13], by refining the criteria proposed there in terms of measurements designed to capture the degree in which a particular creative program involves *fine tuning*, in the sense of tailoring the design of the program to produce a particular kind of output.

To achieve this the following concepts are introduced:

- *Output set* O_K: the set of items produced by a program using knowledge K
- *Re-inventions set* R_K: the set of items already present in the inspiring set reproduced by a program using knowledge K
- *Creative set* C_K: the set of valuable items produced by a program using knowledge K, excluding those in the inspiring set
- *Dependency set* $D_{K'}$ of a subset $K^{'}$ of the input knowledge K is that part of the valuable results which will be missing from the output if $K^{'}$ is removed from K.

For a particularly creatively useful $K^{'}$ - in the sense that removing it from the input knowledge reduces the valuable results - we can say that $K^{'}$ is *fine tuned* if:

$$|D_{K'} \cap R_K| > 0 \text{ and } |D_{K'} \cap C_K| = 0$$

This corresponds to cases where the contribution of $K^{'}$ to high-value output is restricted to replicating elements that were already present in the inspiring set.

When there are at least some high-valued items contributed by $K^{'}$ ($|D_{K'} \cap C_K| > 0$) the following definition gives the measure of how fine-tuned $K^{'}$ is:

$$ft(K^{'}) = \frac{|D_{K'} \cap R_K|}{|D_{K'} \cap C_K|}$$

This returns a value greater than 1 if $K^{'}$ mainly rediscovers already-known artefacts rather than finding new ones of value, and it returns values of 1 or less otherwise.

3 Automatic Generation of Poetry

The automatic generation of poetry has called upon itself a certain amount of interest in the recent past. The complexity of the task, involving several levels of language use (phonetics, lexical choice, syntax, semantics, discourse structuring...) gives rise to a domain of artefacts of high complexity, where a considerable amount of input knowledge is required. The various approaches that have been attempted so far differ considerably in the amount of input knowledge that the creative programs are provided with to carry out their task. Additionally they present important differences in their overall architecture, which allow a rough grouping into: template based, generate and test, evolutionary, and case-based reasoning approaches.

3.1 Template Based Poetry Generation

The ALAMO group [1] has been generating poems in French automatically for some time. Amongst other activities dedicated to the promotion of the use of computers for literary creativity, this group present a number of text generation programs (*literaciels*) which they use for animating literary workshops.

A number of examples of these programs is described in their web site. From the information provided (example of results, brief description of the methods followed) the general idea behind these programs seems to be to identify a set of texts, words , and valid transformations that allow automatic generation of a fixed number of alternative versions of a given poem. For instance they describe the construction method for *Rimbaudelaires*, sonnets obtained by combining the sentence structure of sonnets by Rimbaud with the vocabulary of the poetry of Baudelaire. A poem shell is obtained by cutting out the nouns, verbs, and adjectives from a given sonnet by Rimbaud. Words from the vocabulary of Baudelaire are then used to fill the resulting gaps, following "strong syntactic and rhythmic constraints".

Although their methods seem to be based on identifying basic poem structures which allow a number of variation (in terms of substitution of words for size-matching equivalents at given substitution

hot spots), the resulting effect is striking, and does give the impression of a reasonably articulate poet at work.

How creative this approach can be considered could be discussed. However, the material provided in the ALAMO web page is not enough to apply the type of evaluations described above.

3.2 Generate and Test Approaches

The WASP system [5] draws on prior poems and a selection of vocabulary provided by the user to generate a metrically driven recombination of the given vocabulary according to the line patterns extracted from prior poems. The WASP automatic poet 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.

An initial work by Manurung [9], based on chart generation, focuses on the generation of poetry in English, starting from a semantic representation of the meaning of the desired poem. A very important driving principle in this case is to respect the unity between form and meaning that is considered to provide the aesthetical backbone of real poetry. This implies that poems to be generated must aim for some specific semantic content, however vaguely defined at the start of the composition process.

The approach relied on chart generation, taking as input a specification of the target semantics in first order predicate logic, and a specification of the desired poetic form in terms of metre. Words are chosen from a lexicon that subsume the input semantics, and a chart is produced incrementally to represent the set of possible results. At each stage, the partial solutions are checked semantically to ensure that no sentences incompatible with the original input are produced. Additionally, partial results are checked for compatibility with the desired poetic form. Because the search space is pruned at each stage of invalid partial solutions, the approach is generally efficient.

This approach follows a generate & test approach, in a systematic way, trying all possibilities and making sure that no partial constituent is generated twice by the system It also allows the user to control the input in terms of meaning. This has the advantage of restricting somewhat the probability of obtaining non-sensical output, but it also limits the degree of freedom of the system. The amount of creativity that system can exercise on the semantics of its output is limited.

3.3 Evolutionary Approaches

Levy [8] proposes a computer poet based on evolutionary computation, aided by an evaluation function implemented as a neural network trained on data obtained from human testers.

An important driving principle of this work is to take the real process of human poetry writing as a reference from which to draw the intuitions that drive the system. A very insightful and intuitive algorithmic description of the creative process of poetry composition is provided as a starting point. Based on this description a general architecture for a computer poet is presented. Such an architecture would have a number of generator modules which produce an initial population of candidate poems and modify it in succeeding generations, a number of evaluator modules that select the highest ranking individuals in each generation, a work space in which the current population resides, a lexicon, a conceptual knowledge base, and a syntactical knowledge base. An interesting feature of this architecture is that the evaluators are organised in two tiers - a lower one where actual evaluation of each candidate takes place, and a higher one in charge of focus, which concentrates system effort on the high ranking candidates.

Levy's discussion of the real process of composition suggests there is an interactive dialogue between the top down process of generating drafts to fit a preconceived idea and a bottom up process of accommodating the preconceived idea to fit any appealing partial results that may have been obtained during composition. This is somehow captured by the existence of a threshold of awareness below which draft generation is automatic, and over which striking drafts are specifically chosen to be worked upon.

The Poevolve system is an implemented prototype of this general architecture. The current version operates over a representation of the lexicon that centers on the phonetical information. Evaluation is carried out by a neural network trained on judgements provided by a panel of experts on a single parameter - the likability or creativity of a poem - on a scale of 1 to 6. The system generates randomly an initial population and allows it to evolve by applying a set of operations - mutation, crossover, and direct copy - which in general terms are restricted to substituting one word for another. The results of the prototype are said to seem random in many ways, though there is an increase in value over the course of a program run.

The results of his earlier chart generation approach led Manurung to attempt an evolutionary solution [11, 10]. This system 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.

3.4 Case-Based Reasoning Approaches

An alternative architecture has been attempted which relies on retrieving existing poems similar to a target message provided by the user as a text, and adapting them to fit the required content.

The ASPID system [4] provides 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 establishes a set of priorities over the complete available vocabulary. The next word to be added to the poem draft is 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 have been found in the previous step. This procedure improves search times considerably and it makes 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 fails to ensure successful termination.

ASPERA [6, 7] is a forward reasoning rule-based system that is 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).

4 APPLYING CREATIVITY MEASUREMENTS TO A PARTICULAR EXAMPLE

In order to carry out this experiment results were available for the WASP, ASPID, and ASPERA systems. On first analysis, it was noticed that the data collected for the original evaluations of WASP could be fitted to the scheme proposed by Ritchie. This represented a two-fold advantage.

On one hand data were available with no need for further evaluation processes. The evaluation of poems requires a set of volunteers to read through a set of results producing a quantitative evaluation for each one of the chosen parameters. Such an effort had been carried out for the WASP system [5]. The resulting set of data, in the absence of a methodological framework suited for their analysis, had proved less productive than expected. As a result, the evaluation of subsequent attempts had been more focused on aspects directly relevant to specific design issues [4, 6].

On the other hand, there are further versions of the system that had evolved from the evaluated system. The new versions had been designed based on a simple analysis of the results with no specific methodological framework. This meant that any conclusions obtained by applying the frameworks could be compared with the conclusions found at the time. This should demonstrate whether the application of the proposed method is useful for bringing out informative conclusions from raw data.

We are therefore considering a generating program, WASP, which for the present purposes can be described as follows.

The *inspiring set* is taken to be a specific 16th century Spanish classical sonnet. This establishes a number of restrictions on the poetry that is to be composed. Lines should have 11 syllables, according to very strict stress patterns.

To simplify matters, the artefacts that the program will aim for will be the simpler stanzas that make up a sonnet (two *cuartetos* - four lines each, rhyming ABBA ABBA - and two *tercetos* - three lines each, rhyming either ABA BAB or ABC ABC). As a first approximation, the generating program is set to attempt a *cuarteto* in isolation.

The construction process that is employed is designed to ensure that all resulting items have the correct syllable count and a valid pattern of stressed syllables for each line. Given a specific stanza to aim for, the system attempts to build an instance of this stanza based on the set of line patterns it receives and the available vocabulary. Wherever several possible choices of words match the metric constraints, the program makes a random choice. This provides the nondeterminism required to obtain multiple results on different runs. In each case, the final result may have reached the required number of lines or it may have stopped beforehand - unable to meet the metric constraints on the remaining lines with the material available.

The system is allowed a certain freedom in the following aspects:

- may or may not find rhymes between lines
- may or may not complete a full stanza
- may or may not achieve a syntactically correct poem

4.1 Applying Ritchie's Criteria

Ritchie presents his criteria for assessing creativity based on the assumption that running the program produces a set of basic items, rather than a single item. In this cases, this may be simulated by running the program with the same parameters several times, in order to produce a set of items.

4.1.1 Assessing Creativity based on Existing Evaluation

As a first approach, the evaluation functions *typ* and *val* are defined informally in the following terms. A poem is considered typical if it has the required number of lines and it has a syntactically correct reading. A poem is considered good if anything in it appeals to the aesthetic sense of the evaluator.

The *mapping function* that takes from a particular combination of the initial data values to a specific set of results is given by the construction algorithm (while no full stanza has been achieved, find an appropriate line pattern, and fill that line pattern with adequate words).

The initialisation required to set this process in motion must provide the following information:

- *alternatives for line patterns*: these are obtained from the lines in the sonnet used as inspiring set, and each one corresponds to the sequence of POS tags corresponding to the words appearing in a line of the sonnet
- *alternatives for vocabulary*: these are obtained from the words of the original sonnet plus a number of additional words; each word carries additional information relating to the POS tag corresponding to it, the number of syllables, the position of its stress, and word boundary information that affects the way it combines with neighbouring words to form the metre of the line
- *alternatives for structure*: in the present case, the types of stanza under consideration; a specific stanza must be chosen.

The set of parameters that act as *initial data values* are:

- a set of patterns
- a given vocabulary
- a specific stanza to aim for

Each run of the program with such an initialisation produces either a complete stanza of the desired form or as many lines as can be produced while meeting the metric criteria. In order to obtain results that can be analysed according to Ritchie's framework, each set of 12 runs with the same initialisation is studied as a single set of results. Fourteen different initialisations are considered. This gives a total of 168 resulting poems. Each poem is evaluated by a team of volunteers, who are asked to provide two numerical values: one measuring the syntactic correctness of the poem (on a scale from 0 to 5) and one measuring the aesthetic qualities of the poem (on a similar scale). These values are combined with the number of lines of each poem to provide an approximation to the two evaluation functions required.

The resulting values for these results under Ritchie's criteria are presented in table 2. Table 3 presents the parameters that have been employed to construct the table.

Table 2. Results for 14 criteria

Criterion 1	Average typicallity	0,71
Criterion 2	Typical results / results	0,54
Criterion 3	Average quality	0,47
Criterion 4	Good results / results	0,24
Criterion 5	Good typical results / typical results	0,36
Criterion 6	Good atypical results / results	0,05
Criterion 7	Good atypical results / atypical results	0,12
Criterion 8	Good atypical results / good typical results	0,28
Criterion 9	Results in the inspiring set / inspiring set	0,00
Criterion 10	Results / results in the inspiring set	∞
Criterion 11	Average typicallity new results	0,71
Criterion 12	Average quality new results	0,47
Criterion 13	Typical new results / results	0,54
Criterion 14	Good new results / results	0,24

Only the first eight criteria are relevant, because none of the inspiring set reappears in the result. This is apparent in the fact that criterion 10 tends to infinity as the number of results already present in the inspiring set tends to 0. This is due to the fact that the construction process actually first factorises and then recombines elements of the inspiring set, adding additional words from the vocabulary. This reduces greatly the probability that an element in the inspiring set be generated anew by the system. An immediate consequence is that criterion 9 drops to zero and criterion 10 runs up to infinity. Additionally, those criteria designed to capture specific differences between items that are new and items already in the inspiring set produce the same score as the original criteria they are evolved from (the same values result for criteria 11 and 1, 12 and 3, 13 and 2, 14 and 4).

Table 3. Basic Data for First Approach

Weight for poem length	0,5
Weight for syntactic correctness	0,5
Typicallity threshold	0,7
Quality threshold	0,7
Total number of results	168
Number of Items in the Inspiring Set	2

A question that may need detailed discussion is how one identifies whether an element in the inspiring set is reappearing in the results. For this version of the system, none of the *cuartetos* in the inspiring set appears as such among the results, but some of the lines of the poems in the inspiring set may reappear, and - given the construction procedure employed - all of the lines in the results will have a syntactic structure that is borrowed from the lines in the inspiring set.

The system is better at producing typical items than at producing good items (score higher for criterion 1 than for criterion 3) and higher for criterion 2 than for criterion 4. This makes sense, since all system decisions (algorithms applied and constraints imposed) during the construction process are concerned with ensuring the production of typical items, rather than good ones. In fact, the system has no means for identifying good items, and therefore cannot be expected to aim towards them during construction.

Atypical results score badly in terms of quality. This may be due to evaluators not having a clear idea of whether their judgement on the quality should take into account how typical the item is. Evaluators may be awarding good scores on quality to items that are typical. This would imply that their own reaction is to apply criterion 5 rather than criterion 4. The fact that the system performs better under criterion 5 than criterion 4 with these evaluators may be taken as evidence in favour of this interpretation. Criterion 8 provides an indication of this relation (low presence of atypical results among the good results).

4.1.2 Effect of Evaluation Parameters on Creativity Assessment

The data presented so far are based on a specific selection of parameters to be employed during evaluation. The value obtained for *typ* is actually the result of combining mathematically the values assigned for syntactic correctness and the number of lines obtained for each attempted instance of the stanza. The actual formula applied to obtain the final value corresponds to what Ritchie calls a *weighted property rating scheme*, as used for evaluating typicallity. The role of the weights employed in the actual combination needs to be discussed.

Additionally, two threshold values have been applied to distinguish highly rated items whether on typicallity or quality.

The present section considers whether alternative assignments of values to these parameters affect in any significant way the conclusions drawn on the results. Criteria from 9 to 14 have been omitted from the discussion, since they play no significant role.

The first decision that may affect the evaluation is the relative importance that number of lines and syntactic correctness play in our assessment of typicallity. The evaluation discussed above considers them with equal relative importance: weight assignment for syntactic correctness=0.5 and weight assignment for number of lines=0.5. Table 4 considers a similar evaluation but including two new different alternative weight assignments:

- alternative A0 (the original one with weight for syntactic correctness=0.5 and weight assignment for number of lines=0.5),
- alternative A1 (weight for syntactic correctness=0.7 and weight assignment for number of lines=0.3), and
- alternative A2 (weight for syntactic correctness=0.3 and weight assignment for number of lines=0.7).

Comparatively, alternative A1 gives more importance to syntactic correctness, alternative A0 gives them equal importance, and alternative A2 gives more importance to number of lines.

Table 4. Different weighting for typicallity

Crit.		A0	A1	A2
1	Average typicallity	0,71	0,67	0,75
2	Typical results / results	0,54	0,48	0,79
3	Average quality	0,47	0,47	0,47
4	Good results / results	0,24	0,24	0,24
5	Good typical results / typical results	0,36	0,34	0,29
6	Good atypical results / results	0,05	0,08	0,01
7	Good atypical results / atypical results	0,12	0,16	0,06
8	Good atypical results / good typical results	0,28	0,52	0,05

The results show that average typicallity (criterion 1) drops for alternative A1 and rises for alternative A2. This is due to the fact that the current constraints applied during construction take explicitly into account only the number of lines of the stanza, but correct syntax is only implicitly considered in the reuse of line patterns.

Another possible way of affecting the evaluation is to vary the thresholds that are used to distinguish highly rated items in each class (typical or good). Criteria 1 and 3 are not affected by this change, since they do not refer to the threshold value. Therefore they are omitted from the following discussion.

The threshold value on quality determines how many items are considered good, and therefore affects criteria 4 through to 8. The threshold value on typicallity affects criterion 2 and criteria 5 through to 8.

In the result sets discussed below the weight assignment for typicallity is maintained at syntactic correctness=0.5 and number of lines=0.5 - the same as for the initial discussion.

Table 5 shows the values for the relevant criteria over the same original with five different threshold combinations for quality and typicallity:

- (A) equal high thresholds,
- (B) equal medium thresholds,
- (C) equal low thresholds,
- (D) high typicallity and low quality thresholds, and
- (E) low typicallity and high quality thresholds.

Table 5. Equal high thresholds

Crit.	A	B	C	D	E
2	0,54	0,88	0,89	0,54	0,89
4	0,24	0,50	0,68	0,68	0,24
5	0,36	0,57	0,77	0,89	0,28
6	0,05	0,00	0,00	0,21	0,00
7	0,12	0,00	0,00	0,45	0,00
8	0,28	0,00	0,00	0,44	0,00

It can be seen from the results that lowering the typicallity threshold results in a zero score for criteria 6 to 8. This is because they involve good atypical results. By lowering the typicallity threshold the number of atypical items is reduced, and any reduction brings down the number of good items to be found among them. Criteria 2 (regarding typicallity) and 4 (concerned with quality) are inversely proportional to the threshold applied in each case - the value for the corresponding criteria falls when the threshold rises and falls when it rises. Criteria 5 is different in every case because it involves both thresholds.

There is a great variation between the values obtained for each of these criteria when the thresholds are moved. This implies that the assignment of specific values for these thresholds should be established beforehand based on domain specific criteria, or oriented towards the specific aims that have been established for the system.

An additional alternative is to consider different thresholds for distinguishing typical and atypical items. So far, items that did not rate highly on typicality have been considered atypical. A finer grained approach would establish a low threshold below which items would be considered as atypical. This might establish a high threshold value to determine when an item is typical and a low threshold value to determine when an item is atypical.

4.1.3 Fine Tuning Evaluation

The criteria defined by Colton et al for measuring fine-tuning are somewhat difficult to apply to this case for the following reasons:

- It is difficult to isolate particular items of knowledge from the rest, since the contribution for a given line pattern, for instance, is tightly coupled to the existence of word items of the necessary category
- The same words may be coupled with different line patterns if they belong to a category which appears in more than one

For simplicity, the whole set of knowledge employed is considered as $K^{'}$ for a first approximation of the criteria.

In the case under discussion there are no re-inventions. This means that for the particular set of input knowledge employed, $C_K = O_K$. With respect to the criteria described for measuring the degree of fine tuning $ft(K)$ yields a value of 0 - there are no reinventions among the dependency set for the knowledge K employed.

This measure can be misleading, and it is probably related to having applied too strict an interpretation of what it takes for an artefact to be considered as part of R_K. In their description of the measurements, Colton et al [3] consider only whether an artefact in the output set is exactly the same as one in the inspiring set. This is adequate for the mathematical domain - in which the discussion in that paper is mostly based -, where the elements being generated are structurally simple and the probability of replicating the ones in the inspiring set is high. In the domain of poetry, as soon as the conceptual unit of the poem is broken down into its constituent elements and the construction process is allowed to recombine these elements in different ways, the probability of reproducing a poem originally in the inspiring set is very low.

For these cases, it may be more fruitful to employ some measure of similarity between the artefacts in the output set and those in the inspiring set to decide which artefacts are to be considered as 're-inventions'. Artefacts that are very similar to those in the inspiring set, even if they are different, should be considered as re-inventions. A creative program that produces artefacts only marginally different from those in the inspiring set should probably be considered to be using fine-tuned knowledge.

In this sense, the criteria proposed for measuring fine-tuning should be extended for the case of artefact domains of higher complexity, possibly taking into account measures akin to those proposed in Pease et al [12].

4.2 Analysis of the Results

The type of evaluation employed seems to fall short in terms of identifying the real value of the resulting poems. This may be due to unforeseen assumptions on the part of the human evaluators about what is considered novelty and quality in this context.

We consider that typical *cuartetos* have four lines. Results with less than four lines will be considered atypical - the shorter the more atypical.

Typical *cuartetos* in 16th century Spanish poetry generally fulfil the following conditions:

- They include striking vocabulary items (words still in use but then used with different meaning, words no longer used, words referring to objects or concepts that date specifically from that time)
- Their grammar is sometimes difficult to understand (due to obsolete turns of phrase or the use of *hyperbaton*, a poetic ornament which relies on shuffling the elements of a sentence in ungrammatical ways in order to satisfy metric constraints or achieve poetic effects)
- They seldom occur in isolation, so they are rarely self contained units from a syntactic or semantic point of view

These issues may have played a role in making evaluators rate highly some of the resulting poems: presence of striking vocabu-

lary items, obscure grammar, attribution of a hypothetical context in which certain turns of phrase might make sense. However, they are not necessarily desirable features in poems of a wider domain.

In view of this conclusion, a more restricted method of evaluation should be defined to represent more closely the ingredients at play in this domain. To counter the effect described, such a method of evaluation should provide the evaluators with explicit guidelines on how to rate the poems, and which features to take into account when doing so. For instance, it might be considered that a *cuarteto* is good depending on the following aspects:

- Rhyme (very good if it rhymes ABBA, good if all verses rhyme in some way, acceptable if some verses do not rhyme, and bad if none of the verses rhyme)
- Syntax (very good if it is a self contained syntactically correct unit, good if it can be parsed as a syntactically correct fragment within some hypothetical context, acceptable if it does not contain any striking syntactical errors, and bad if it does)
- Poetic ornaments (the quality of a poem rises if it contains any combination of words that can be interpreted as a poetic ornament)

5 DISCUSSING EVALUATION FOR DIFFERENT ARCHITECTURES

The different architectures for the automatic generation of poetry that were outlined in section 3 give rise to different issues that may have to be taken into account when designing adequate evaluation methods.

5.1 Template Based

The method of production described for Rimbaudelaires differs from the other approaches described in that it is template based, in the sense that there is a husk of the poem which contains a number of words already in fixed positions, and which cannot be modified by the generation process. The degree of freedom involved is very low. In contrast, all other systems allow modification at word level on all positions of the poem. Manurung's evolutionary system in particular actually allows manipulation at all levels of representation.

The poetry produced by the ALAMO group should rate highly on typicality and quality, but low on originality. This is a particularly good case for applying the criteria related to a comparative study of how much of the result set is included in the inspiring set. It is possible that all the result set for systems based on this architecture be present in the inspiring set from the start.

5.2 Generate & Test

The two initial systems presented by Gervás [4, 5] 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. Regarding evaluation, such systems would require a definition of the input knowledge in terms of the set of line patterns being considered, and the additional vocabulary. The inspiring set in each case is explicitly defined as the set of original poems from which the line patterns are drawn. It would still have to be decided whether the introduction of additional vocabulary outside the words appearing in those poems should be considered as an extension of the inspiring set. Maybe the concept of inspiring set should be redefined to include this sort of situation.

In contrast, the chart generation work of Manurung [9] employs much more complex input knowledge, regarding syntax, semantics, and metre. In this case it is not so clear what the inspiring set can be considered, because no poems are mentioned explicitly as being taken by the program as inspiration or input knowledge, but rather the restrictions for a particular poetic form are employed.

5.3 Evolutionary

The evaluation that has been applied to the WASP system might be extended to those following the evolutionary approach. However, certain differences between the two approaches must be taken into account.

Poevolve and WASP rely on a different high level description of the process of composition. Poevolve relies on a generate, evaluate, evolve, cycle, whereas WASP follows s simple generate & test method. Of these, the description on which Poevolve is based is possibly a more accurate description of the actual creative process. Given enough information, (on the elements that are manipulated and in such a way that the evaluators can take it into account), it does have a great potential as claimed. However, it is not clear whether the amount (or rather the kind) of information required (syntactic, semantic) is as easily coded in terms of connectionist computing as the kind of information that the current prototype of Poevolve is using (mostly phonology and metrics).

On the other hand the first prototype of Poevolve and WASP have in common the underlying assumption that consideration of phonology and metrics with little regard for semantics and syntax does lead to reasonably 'poem-like' results. This may be related to the discussion in [11] regarding how automatically generated of poetry is evaluated by humans with more leniency than the equivalent efforts in prose, and how this holds a danger of relaxing into easy simulations with little real merit.

The use of a neural network in Poevolve to evaluate the results could solve many of the problems faced when evaluating automatically generated poems. Nonetheless, the introduction of the training process for a neural net within the evaluation/feedback loop applied by the program introduces additional complexity in the already nebulous chain of valuations that take place when humans judge poetry.

Again, when defining evaluation methods for the evolutionary system of Manurung [11, 10] it will be difficult to consider whether there is such a thing as an inspiring set, since the system seems to be working rather from general rules about poetry than specific examples of poems.

5.4 Case Based Reasoning

The ASPID and ASPERA systems are evolved versions of the WASP system that employ CBR techniques. From the CBR point of view the main difference between them is that ASPID operates on line-sized cases which it composes to build coherent stanzas, and ASPERA operates with stanza-sized cases. Additionally, ASPERA is a more complex system, having extra modules for user interaction. These modules request some additional input parameters from the user concerning the setting, mood and length of the intended message, and apply a knowledge based system to filter an appropriate starting set of cases and vocabulary.

In general terms, the construction modules of both systems start from an initial target content provided by the user (intended message), and retrieve a case to later use the solution as seed structure which to fill in. The intended message is used as main source for

the words required to fill in the case, and the case itself as default source. An additional vocabulary (also provided by the user) is used as intermediate source.

Regarding the kind of evaluation discussed here, there is one particular issue that needs to be taken into account. The intended message provided by the user, by allowing the user some control over the ingredients that will be used to produce the final result, may affect the question of whether the system is fine-tuned. By exercising this control this to guide the system towards particular results, the system can be forced to produce results that are very close to the inspiring set.

In the case of ASPERA, further control features are provided in the form of initial basic data that act as system parameters. These control the way in which the system splits the target content over the number of lines in the chosen stanza, the kind of similarity employed by the system to retrieve cases which to use as seed for the construction process, the number of syllables required per line, the number of lines in the stanza, and the amount of variation in relative position of a word between the target content and the final result. They are used to provide the system with a certain degree of freedom which can be controlled by the user. This feature results in a system that can be either configured to generate conservative versions that replicate an important portion of the inspiring set, or innovative versions that depart from the inspiring set.

The analysis on fine-tuning in Colton et al [3] presents the concept as a property exhibited by some creative systems, inherent to their design, and indicative of a certain lack of general applicability. The described mechanisms in ASPID and ASPERA provide the means for the user to control the degree in which the system will try to reproduce its inspiring set. In a way, the criteria proposed by Colton could be applied to obtain a measure of the extent in which these features affect the relationship between inspiring set and result.

6 CONCLUSIONS

The evaluation of automatically generated poetry still requires a lot of work. Existing theoretical proposal on how to evaluate creativity can be applied to this domain, but fall short in various ways due to the intrinsic complexity of the artefacts being generated and the kind of input knowledge that is needed.

In general terms, the aspects that need evaluation can be summarised as follows. Artefacts must at the same time be capable of:

- Meeting a subset M of the requirements of the evaluation function (*met requirements*)
- Challenging a subset C of the requirement of the evaluation function (*challenged requirements*)

An existing set of artefacts of a given kind allows the prediction of subsequent artefacts of the same kind, with a given probability. Items that are predictable in this sense are not considered creative. Items that cannot be adequately linked to the preceding sequence are not considered creative.

This issue must be discussed relative to the particular domain to which the items belong. For conservative domains, typical items tend to be good, and good items tend to be typical. This approach gives rise to a somewhat stilted style. For innovative domains atypical items tend to be good, and typical items are bad. The resulting style is more dynamic.

Relative to Boden's distinction between exploratory and transformational creativity, conservative domains rely more on exploratory creativity and innovative domains rely more on transformational creativity.

For the particular field of poetry, as considered in existing attempts at automatic generation, the identification of the domain in the sense used here relates more to the poetic form that is being aimed for. Certain poetic forms are more conservative, and others are more innovative.

Regarding the issue of measuring how fine-tuned a creative program is to produce a particular type of items, existing descriptions of how this measurement may be achieved need to be extended to take into account domains where artefacts that are too similar to those in the inspiring set may be considered just as un-original as the items in the inspiring set. Such is the case in the poetry domain discussed in this paper.

On the other hand, measuring fine-tuning may not be enough to give an idea of how adequate a creative program is, in the sense that an equivalent measure may be required to capture possible inadequacies at the other extreme: programs that are unable to reproduce any of their inspiring set.

ACKNOWLEDGEMENTS

I would like to thank the group of volunteers who carried out the task of assigning real values to the set of results - poems - considered in this paper, and to the referees for their comments which helped improve this paper.

REFERENCES

[1] ALAMO. Atelier de Littérature Assisté par la Mathématique et les Ordinateurs. http://indy.culture.fr/alamo/rialt/pagaccalam.html, 2000.
[2] M.A. Boden, *The Creative Mind*, Weidenfeld and Nicholson, London, 1990.
[3] S. Colton, A. Pease, and G. Ritchie, 'The effect of input knowledge on creativity', in *Proc. of the ICCBR-01 Workshop on Creative Systems*, (2001).
[4] P. Gervás, 'Un modelo computacional para la generación automática de poesía formal en castellano', *Procesamiento de lenguaje natural*, **26**(26), (2000).
[5] P. 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*, pp. 93–100, (2000).
[6] P. Gervás, 'Creativity versus faithfulness', in *Proc. of the AISB-01 Symposium on AI and Creativity in Arts and Science*, (2001).
[7] P. Gervás, 'An expert system for the composition of formal Spanish poetry', *Journal of Knowledge-Based Systems*, **14**(3–4), 181–188, (2001).
[8] R. P. Levy, 'A computational model of poetic creativity with neural network as measure of adaptive fitness', in *Proc. of the ICCBR-01 Workshop on Creative Systems*, (2001).
[9] H. M. Manurung, 'Chart generation of rhythm-patterned text', in *Proc. of the First International Workshop on Literature in Cognition and Computers, Tokyo, 1999*, (1999).
[10] 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).
[11] H. M. Manurung, G. Ritchie, and H. Thompson, 'Towards a computational model of poetry generation', in *Proc. of the AISB-00 Symposium on Creative & Cultural Aspects of AI*, (2001).
[12] A. Pease, D. Winterstein, and S. Colton, 'Evaluating machine creativity', in *Proc. of the ICCBR-01 Workshop on Creative Systems*, (2001).
[13] G. Ritchie, 'Assessing creativity', in *Proc. of the AISB-01 Symposium on AI and Creativity in Arts and Science*, (2001).