Empirically-based Control of Natural Language Generation

Daniel S. Paiva
Department of Informatics
University of Sussex
Brighton, UK
danielpa@sussex.ac.uk

Roger Evans
Information Technology Research Institute
University of Brighton
Brighton, UK
Roger.Evans@itri.brighton.ac.uk

Abstract

In this paper we present a new approach to controlling the behaviour of a natural language generation system by correlating internal decisions taken during free generation of a wide range of texts with the surface stylistic characteristics of the resulting outputs, and using the correlation to control the generator. This contrasts with the generate-and-test architecture adopted by most previous empirically-based generation approaches, offering a more efficient, generic and holistic method of generator control. We illustrate the approach by describing a system in which stylistic variation (in the sense of Biber (1988)) can be effectively controlled during the generation of short medical information texts.

1 Introduction

This paper[1] is concerned with the problem of controlling the output of natural language generation (NLG) systems. In many application scenarios the generator's task is underspecified, resulting in multiple possible solutions (texts expressing the desired content), all equally good to the generator, but not equally appropriate for the application. Customising the generator directly to overcome this generally leads to ad-hoc, non-reusable solutions. A more modular approach is a generate-and-test architecture, in which all solutions are generated, and then ranked or otherwise selected according to their appropriateness in a separate post-process. Such architectures have been particularly prominent in the recent development of empirically-based approaches to NLG, where generator outputs can be selected according to application requirements acquired directly from human subjects (e.g. Walker *et al.* (2002)) or statistically from a corpus (e.g. Langkilde-Geary (2002)). However, this approach suffers from a number of drawbacks:

1. It requires generation of all, or at least many solutions (often hundreds of thousands), expensive both in time and space, and liable to lead to unnecessary interactions with other components (e.g. knowledge bases) in complex systems. Recent advances in the use of packed representations ameliorate some of these issues, but the basic need to compare a large number of solutions in order to rank them remains.
2. The 'test' component generally does not give fine-grained control — for example, in a statistically-based system it typically measures how close a text is to some single notion of ideal (actually, statistically average) output.
3. Use of an external filter does not combine well with any control mechanisms *within* the generator: e.g. controlling combinatorial explosion of modifier attachment or adjective order.

In this paper we present an empirically-based method for controlling a generator which overcomes these deficiencies. It controls the generator internally, so that it can produce just one (locally) optimal solution; it employs a model of language *variation*, so that the generator can be controlled within a multidimensional space of possible variants; its view of the generator is completely holistic, so that it can accommodate any other control mechanisms intrinsic to the generation task.

[1] Paiva and Evans (2004) provides an overview of our framework and detailed comparison with previous approaches to stylistic control (like Hovy (1988), Green and DiMarco (1993) and Langkilde-Geary (2002)). This paper provides a more detailed account of the system and reports additional experimental results.

Proceedings of the 43rd Annual Meeting of the ACL, pages 58–65,
Ann Arbor, June 2005.

To illustrate our approach we describe a system for controlling 'style' in the sense of Biber (1988) during the generation of short texts giving instructions about doses of medicine. The paper continues as follows. In §2 we describe our overall approach. We then present the implemented system (§3) and report on our experimental evaluation (§4). We end with a discussion of conclusions and future directions (§5).

2 Overview of the Approach

Our overall approach has two phases: (1) offline calculation of the control parameters, and (2) online application to generation. In the first phase we determine a set of *correlation equations*, which capture the relationship between surface linguistic features of generated texts and the internal generator decisions that gave rise to those texts (see figure 1). In the second phase, these correlations are used to guide the generator to produce texts with particular surface feature characteristics (see figure 2).

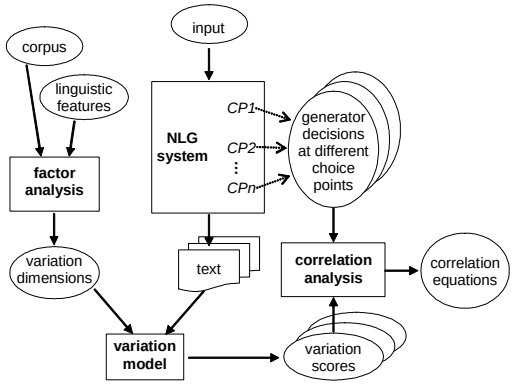

Figure 1: Offline processing

The starting point is a corpus of texts which represents all the variability that we wish to capture. Counts for (surface) linguistic features from the texts in the corpus are obtained, and a factor analysis is used to establish dimensions of variation in terms of these counts: each dimension is defined by a weighted sum of scores for particular features, and factor analysis determines the combination that best accounts for the variability across the whole corpus. This provides a *language variation model* which can be used to score a new text along each of the identified dimensions, that is, to locate the text in the variation space determined by the corpus.

The next step is to take a generator which can generate across the range of variation in the corpus, and identify within it the key choice points (CP_1, CP_2, … CP_n) in its generation of a text. We then allow the generator to freely generate all possible texts from one or more inputs. For each text so generated we record (a) the text's score according to the variation model and (b) the set of decisions made at each of the selected choice points in the generator. Finally, for a random sample of the generated texts, a statistical correlation analysis is undertaken between the scores and the corresponding generator decisions, resulting in correlation equations which predict likely variation scores from generator decisions.

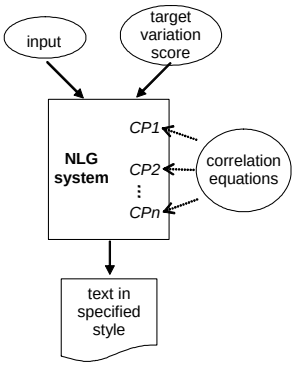

Figure 2: Online processing

In the second phase, the generator is adapted to use the correlation equations to conduct a best-first search of the generation space. As well as the usual input, the generator is supplied with target scores for each dimension of variation. At each choice point, the correlation equations are used to predict which choice is most likely to move closer to the target score for the final text.

This basic architecture makes no commitment to what is meant by 'variation', 'linguistic features', 'generator choice points', or even 'NLG system'. The key ideas are that a statistical analysis of surface features of a corpus of texts can be used to define a model of variation; this model can then be used to control a generator; and the model can also be used to evaluate the generator's performance. In the next section we describe a concrete instantiation of this architecture, in which 'variation' is stylistic variation as characterised by a collection of shallow lexical and syntactic features.

3 An Implemented System

In order to evaluate the effectiveness of this general approach, we implemented a system which attempts to control style of text generated as de-

fined by Biber (1988) in short text (typically 2-3 sentences) describing medicine dosage instructions.

3.1 Factor Analysis

Biber characterised style in terms of very shallow linguistic features, such as presence of pronouns, auxiliaries, passives etc. By using factor analysis techniques he was able to determine complex correlations between the occurrence and non-occurrence of such features in text, which he used to characterise different styles of text.[2]

We adopted the same basic methodology, applied to a smaller more consistent corpus of just over 300 texts taken from proprietary patient information leaflets. Starting with around 70 surface linguistic features as variables, our factor analysis yielded two main factors (each containing linguistic features grouped in positive and negative correlated subgroups) which we used as our dimensions of variation. We interpreted these dimensions as follows (this is a subjective process — factor analysis does not itself provide any interpretation of factors): dimension 1 ranges from texts that try to involve the reader (high positive score) to text that try to be distant from the reader (high negative score); dimension 2 ranges from texts with more pronominal reference and a higher proportion of certain verbal forms (high positive score) to text that use full nominal reference (high negative score).[3]

3.2 Generator Architecture

The generator was constructed from a mixture of existing components and new implementation, using a fairly standard overall architecture as shown in figure 3. Here, dotted lines show the control flow and the straight lines show data flow — the choice point annotations are described below.

The *input constructor* takes an input specification and, using a background database of medicine information, creates a network of concepts and relations (see figure 4) using a schema-based approach (McKeown, 1985).

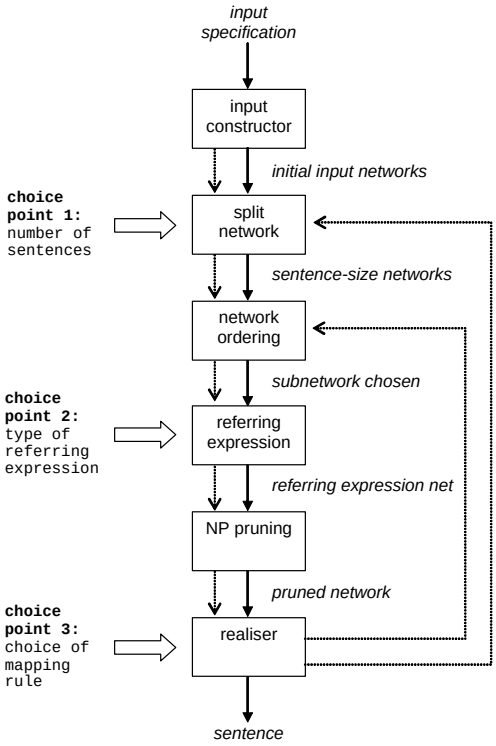

Figure 3: Generator architecture with choice points

Each network is then split into subnetworks by the *split network* module. This partitions the network by locating 'proposition' objects (marked with a double-lined box in figure 4) which have no parent and tracing the subnetwork reachable from each one. We call these subnetworks *propnets*. In figure 4, there are two propnets, rooted in [1:take] and [9:state] — proposition [15:state] is not a root as it can be reached from [1:take]. A list of all possible groupings of these propnets is obtained[4], and one of the possible combinations is passed to the network ordering module. This is the first source of non-determinism in our system, marked as *choice point one* in figure 3. A combination of subnetworks will be material for the realisation of one paragraph and each subnetwork will be realised as one sentence.

[2] Some authors (e.g. Lee (1999)) have criticised Biber for making assumptions about the validity and generalisability of his approach to English language as a whole. Here, however, we use his methodology to characterise whatever variation exists without needing to make any broader claims.

[3] Full details of the factor analysis can be found in (Paiva 2000).

[4] For instance, with three propnets (A, B and C) the list of combinations would be [(A,B,C), (A,BC), (AB, C), (AC,B), (ABC)].

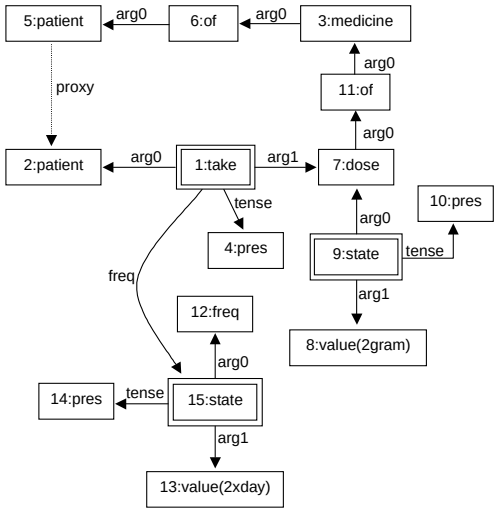

Figure 4: Example of semantic network produced by the input constructor[5]

The *network ordering* module receives a combination of subnetworks and orders them based on the number of common elements between each subnetwork. The strategy is to try to maximise the possibility of having a smooth transition from one sentence to the next in accordance with Centering Theory (Grosz *et al*., 1995), and so increase the possibility of having a pronoun generated.

The *referring expression* module receives one subnetwork at a time and decides, for each object that is of type [thing], which type of referring expression will be generated. The module is re-used from the Riches system (Cahill *et al*., 2001) and it generates either a definite description or a pronoun. This is the second source of non-determinism in our system, marked as *choice point two* in figure 3. Referring expression decisions are recorded by introducing additional nodes into the network, as shown for example in figure 5 (a fragment of the network in figure 4, with the additional nodes).

NP pruning is responsible for erasing from a referring expression subnetwork all the nodes that can be transitively reached from a node marked to be pronominalised. This prevents the realiser from trying to express the information twice. In figure 5, [7:dose] is marked to be pronominalised, so the concepts [11:of] and [3:medicine] do not need to be realised, so they are pruned.

[5] Although some of the labels in this figure look like words, they bear no direct relation to words in the surface text — for example, 'of' may be realised as a genitive construction or a possessive.

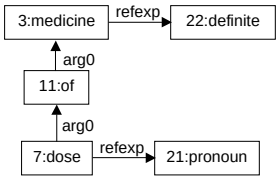

Figure 5: Referring expressions and pruning

The *realiser* is a re-implementation of Nicolov's (1999) generator, extended to use the wide-coverage lexicalised grammar developed in the LEXSYS project (Carroll *et al*., 2000), with further semantic extensions for the present system. It selects grammar rules by matching their semantic patterns to subnetworks of the input, and tries to generate a sentence consuming the whole input. In general there are several rules linking each piece of semantics to its possible realisation, so this is our third, and most prolific, source of non-determinism in the architecture, marked as *choice point three* in figure 3.

A few examples of outputs for the input represented in figure 4 are:

> the dose of the patient 's medicine is taken twice a day. it is two grams.
>
> the two-gram dose of the patient 's medicine is taken twice a day.
>
> the patient takes the two-gram dose of the patient 's medicine twice a day.

From a typical input corresponding to 2-3 sentences, this generator will generate over a 1000 different texts.

3.3 Tracing Generator Behaviour

In order to control the generator's behaviour we first allow it to run freely, recording a 'trace' of the decisions it makes at each choice point during the production of each text. Although there are only three choice points in figure 3, the control structure included two loops: an outer loop which ranges over the sequence of propnets, generating a sentence for each one, and an inner loop which ranges over subnetworks of a propnet as realisation rules are chosen. So the decision structure for even a small text may be quite complex.

In the experiments reported here, the trace of the generation process is simply a record of the number of times each decision (choice point, and what choice was made) occurred. Paiva (2004) discusses more complex tracing models, where the context of each decision (for example, what the preceding decision was) is recorded and used in the correlation. However the best results were obtained using

just the simple decision-counting model (perhaps in part due to data sparseness for more complex models).

3.4 Correlating Decisions with Text Features

By allowing the generator to freely generate all possible output from a single input, we recorded a set of <trace, text> pairs ranging across the full variation space. From these pairs we derived corresponding <decision-count, factor-score> pairs, to which we applied a very simple correlational technique, *multivariate linear regression analysis*, which is used to find an estimator function for a linear relationship (i.e., one that can be approximated by a straight line) from the data available for several variables (Weisberg, 1985). In our case we want to predict the value for a score in a stylistic dimension (SS_i) based on a configuration of generator decisions (GD_j) as seen in equation 1.

(eq. 1) $SS_i = x_0 + x_1GD_1 + \ldots + x_nGD_n + \varepsilon$ [6]

We used three randomly sampled data sets of 1400, 1400 and 5000 observations obtained from a potential base of about 1,400,000 different texts that could be produced by our generator from a single input. With each sample, we obtained a regression equation for each stylistic dimension separately. In the next subsections we will present the final results for each of the dimensions separately.

Regression on Stylistic Dimension 1

For the regression model on the first stylistic dimension (SS1), the generator decisions that were used in the regression analysis[7] are: imperative with one object sentences (IMP_VNP), V_NP_PP agentless passive sentences (PAS_VNPP), V_NP bypassives (BYPAS_VN), and N_PP clauses (NPP) and these are all decisions that happen in the realiser, i.e., at the third choice point in the architecture. This resulted in the regression equation shown in equation 2.

(eq. 2)

SS1 = 6.459 – (1.460∗NPP) – (1.273*BYPAS_VN) – (1.826∗PAS_VNPP) + (1.200∗IMP_VNP)[8]

The coefficients for the regression on SS1 are unstandardised coefficients, i.e. the ones that are used when dealing with raw counts for the generator decisions.

The coefficient of determination (R^2), which measures the proportion of the variance of the dependent variable about its mean that is explained by the independent variables, had a reasonably high value (.895)[9] and the analysis of variance obtained an *F* test of 1701.495.

One of the assumptions that this technique assumes is the linearity of the relation between the dependent and the independent variables (i.e., in our case, between the stylistic scores in a dimension and the generator decisions). The analysis of the residuals resulted in a graph that had some problems but that resembled a normal graph (see (Paiva, 2004) for more details).

Regression on Stylistic Dimension 2

For the regression model on the second stylistic dimension (SS2) the variables that we used were: the number of times a network was split (SPLITNET), generation of a pronoun (RE_PRON), auxiliary verb (VAUX), noun with determiner (NOUN), transitive verb (VNP), and agentless passive (PAS_VNP) — the first type of decision happens in the split network module (our first choice point); the second, in the referring expression module (second choice point); and the rest in the realiser (third choice point).

The main results for this model are as follows: the coefficient of determination (R^2) was .959 and the analysis of variance obtained an *F* test of 2298.519. The unstandardised regression coefficients for this model can be seen in eq. 3.

(eq. 3)

SS2 = – 27.208 – (1.530∗VNP) + (2.002∗RE_PRON) – (.547∗NOUN) + (.356∗VAUX) + (.860∗SPLITNET) + (.213∗PAS_VNP)[10]

[6] SS_i represents a stylistic score and is the *dependent variable* or *criterion* in the regression analysis; the GD_j's represent generator decisions and are called the *independent variables* or *predictors*; the x_j's are weights, and ε is the error.

[7] The process of determining the regression takes care of eliminating the variables (i.e. generator decisions) that are not useful to estimate the stylistic dimensions.

[8] This specific equation came from the sample with 5,000 observations — the equations obtained from the other samples are very similar to this one.

[9] All the statistical results presented in this paper are significant at the 0.01 level (two-tailed).

[10] This specific equation comes from one of the samples of 1,400 observations.

With this second model we did not find any problems with the linearity assumptions as the analysis of the residuals gave a normal graph.

4 Controlling the Generator

These regression equations characterise the way in which generator decisions influence the final style of the text (as measured by the stylistic factors). In order to control the generator, the user specifies a target stylistic score for each dimension of the text to be generated. At each choice point during generation, all possible decisions are collected in a list and the regression equations are used to order them. The equations allow us to estimate the subsequent values of SS1 and SS2 for each of the possible decisions, and the decisions are ordered according to the distance of the resulting scores from the target scores — the closer the score, the better the decision.

Hence the search algorithm that we are using here is the *best-first search*, i.e., the best local solution according to an evaluation function (which in this case is the Euclidian distance from the target and the resulted value obtained by using the regression equation) is tried first but all the other local solutions are kept in order so backtracking is possible.

In this paper we report on tests of two internal aspects of the system[11]. First we wish to know how good the generator is at hitting a user-specified target — i.e., how close are the scores given by the regression equations for the first text generated to the user's input target scores. Second, we wish to know how good the regression equation scores are at modelling the original stylistic factors — i.e., we want to compare the regression scores of an output text with the factor analysis scores. We address these questions across the whole of the two-dimensional stylistic space, by specifying a rectangular grid of scores spanning the whole space, and asking the generator to produce texts for each grid point from the same semantic input specification.

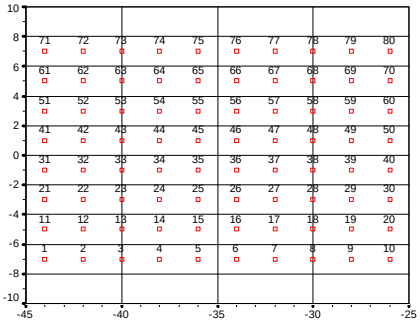

Figure 6: Target scores for the texts

In this case we divided the scoring space with an 8 by 10 grid pattern as shown in figure 6.[12] Each point specifies the target scores for each text that should be generated (the number next to each point is an identifier of each text). For instance, text number 1 was targeted at coordinate (−7, −44), whereas text number 79 was targeted at coordinate (+7, −28).

4.1 Comparing Target Points and Regression Scores

In the first part of this experiment we wanted to know how close to the user-specified target coordinates the resulting regression scores of the first generated text were. This can be done in two different ways. The first is to plot the resulting regression scores (see figure 7) and visually check if it mirrors the grid-shape pattern of the target points (figure 6) — this can be done by inspecting the text identifiers[13]. This can be a bit misleading because there will always be variation around the target point that was supposed to be achieved (i.e., there is a margin for error) and this can blur the comparison unfavourably.

[11] We are not dealing with external (user) evaluation of the system and of the stylistic dimensions we obtained — this was left for future work. Nonetheless, Sigley (1997) showed that the dimensions obtained with factor analysis and people's perception have a high correlation.

[12] The range for each scale comes from the maximum and minimum values for the factors obtained in the samples of generated texts.

[13] Note that some texts obtained the same regression score and, in the statistical package, only one was numbered. Those instances are: 1 and 7; 18 and 24; 22 and 28.

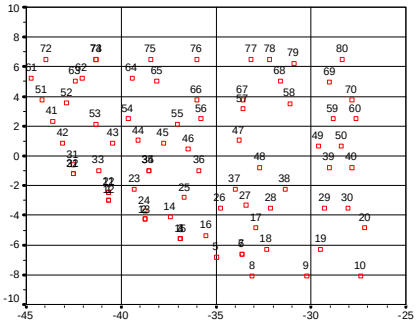

Figure 7: Texts scored by using the regression equation

A more formal comparison can be made by plotting the target points versus the regression results for each dimension separately and obtaining a correlation measure between these values. These correlations are shown in figure 8 for SS1 (left) and SS2 (right). The degree of correlation (R^2) between the values of target and regression points is 0.9574 for SS1 and 0.942 for SS2, which means that the search mechanism is working very satisfactorily on both dimensions.[14]

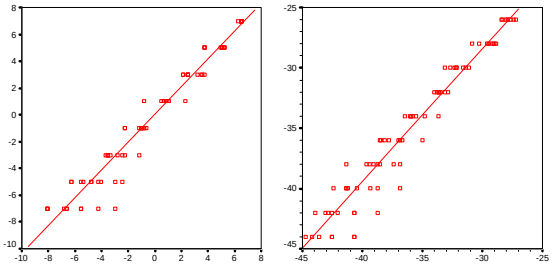

Figure 8: Plotting target points versus regression results on SS1 (left) and SS2 (right)

4.2 Comparing Target Points and Stylistic Scores

In the second part of this experiment we wanted to know whether the regression equations were doing the job they were supposed to do by comparing the regression scores with stylistic scores obtained (from the factor analysis) for each of the generated texts. In figure 9 we plotted the texts in a graph in accordance with their stylistic scores (once again, some texts occupy the same point so they do not appear).

[14] All the correlational figures (R^2) presented for this experiment are significant at the 0.01 level (two-tailed).

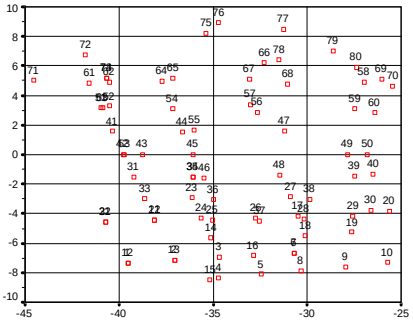

Figure 9: Texts scored using the two stylistic dimension obtained in our factor analysis

In the ideal situation, the generator would have produced texts with the perfect regression scores and they would be identical to the stylistic scores, so the graph in the figure 9 would be like a grid-shape one as in figure 6. However we have already seen in figure 7, that this is not the case for the relation between the target coordinates and the regression scores. So we did not expect the plot of stylistic scores 1 (SS1) against stylistic scores 2 (SS2) to be a perfect grid.

Figure 10 (left-hand side) shows the relation between the target points and the scores obtained from the original factor equation of SS1. The value of R^2, which represents their correlation, is high (0.9458), considering that this represents the possible accumulation of errors of two stages: from the target to the regression scores, and then from the regression to the actual factor scores. On the right of figure 10 we can see the plotting of the target points and their respective factor scores on SS2. The correlation obtained is also reasonably high (R^2 = 0.9109).

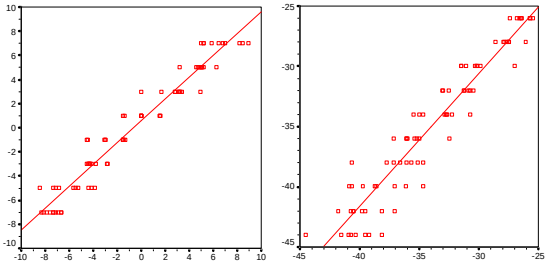

Figure 10: Plotting target points versus factor scores on SS1 (left) and SS2 (right)

5 Discussion and Future Work

These results demonstrate that it is possible to provide effective control of a generator correlating internal generator behaviour with characteristics of the resulting texts. It is important to note that these

two sets of variables (generator decision and surface features) are in principle quite independent of each other. Although in some cases there are strong correlations (for example, the generator's use of a 'passive' rule, correlates with the occurrence of passive participles in the text), in others the relationship is much less direct (for example, the choice of how many subnetworks to split a network into, i.e., SPLITNET, does not correspond to any feature in the factor analysis), and the way individual features combine into significant factors may be quite different.

Another feature of our approach is that we do not assume some pre-defined notion of parameters of variation – variation is characterised completely by a corpus (in contrast to approaches which use a corpus to characterise a *single* style). The disadvantage of this is that variation is not grounded in some 'intuitive' notion of style: the interpretation of the stylistic dimensions is subjective and tentative. However, as no comprehensive computationally realisable theory of style yet exists, we believe that this approach has considerable promise for practical, empirically-based stylistic control.

The results reported here also make us think that a possible avenue for future work is to explore the issue of what types of problems the generalisation induced by our framework (which will be discussed below) can be applied to. This paper dealt with an application to stylistic variation but, in theory, the approach can be applied to any kind of process to which there is a sorting function that can impose an order, using *a measurable scale* (e.g., ranking), onto the outputs of another process.

Schematically the approach can be abstracted to any sort of problem of the form shown in figure 11. Here there is a *producer* process outputting a large number of solutions. There is also a *sorter* process which will classify those solutions in a certain order. The numerical value associated with the output by the *sorter* can be correlated with the decisions the producer took to generate the output. The same correlation and control mechanism used in this paper can be introduced in the producer process, making it controllable with respect to the sorting dimension.

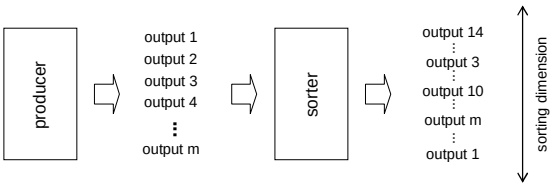

Figure 11: The producer-sorter scheme.

References

Biber, Douglas (1988) *Variation across speech and writing*. Cambridge University Press.

Cahill, Lynne; J. Carroll; R. Evans; D. Paiva; R. Power; D. Scott; and K. van Deemter From RAGS to RICHES: exploiting the potential of a flexible generation architecture. *Proceedings of ACL/EACL 2001*, pp. 98-105.

Carroll, John; N. Nicolov; O. Shaumyan; M. Smets; and D. Weir (2000) Engineering a wide-coverage lexicalized grammar. *Proceedings of the Fifth International Workshop on Tree Adjoining Grammars and Related Frameworks.*

Green, Stephen J.; and C. DiMarco (1993) Stylistic decision-making in NLG. In *Proceedings of the 4th European Workshop on Natural Language Generation*. Pisa, Italy.

Grosz, Barbara J.; A.K. Joshi; and S. Weinstein (1995) *Centering: A Framework for Modelling the Local Coherence of Discourse*. Institute for Research in Cognitive Science, IRCS-95-01, University of Pennsylvania.

Hovy, Eduard H. (1988) *Generating natural language under pragmatic constraints*. Lawrence Erlbaum Associates.

Langkilde-Geary, Irene. (2002) An empirical verification of coverage and correctness for a general-purpose sentence generator. *Proceeding of INLG'02*, pp. 17-24.

Lee, David (1999) *Modelling Variation in Spoken And Written English: the Multi-Dimensional Approach Revisited*. PhD thesis, University of Lancaster, UK.

McKeown, Kathleen R. (1985) *Text Generation: Using Discourse Strategies and Focus Constraints to Generate Natural Language Text*. Cambridge University Press.

Nicolov, Nicolas (1999) *Approximate Text Generation from Non-hierarchical Representations in a Declarative Framework*. PhD Thesis, University of Edinburgh.

Paiva, Daniel S. (2000) Investigating style in a corpus of pharmaceutical leaflets: results of a factor analysis. *Proceedings of the Student Workshop of the 38th Annual Meeting of the Association for Computational Linguistics (ACL'2000)*, Hong Kong, China.

Paiva, Daniel S. (2004) *Using Stylistic Parameters to Control a Natural Language Generation System*. PhD Thesis, University of Brighton, Brighton, UK.

Paiva, Daniel S.; R. Evans (2004) A Framework for Stylistically Controlled Generation. In *Proceedings of the 3rd International Conference on Natural Language Generation (INLG'04)*. New Forest, UK.

Sigley, Robert (1997) Text categories and where you can stick them: a crude formality index. *International Journal of Corpus Linguistics*, volume 2, number 2, pp. 199-237.

Walker, Marilyn; O. Rambow, and M. Rogati (2002) Training a Sentence Planner for Spoken Dialogue Using Boosting. *Computer Speech and Language, Special Issue on Spoken Language Generation*. July.

Weisberg, Sanford (1985) *Applied Linear Regression*, 2nd edition. John Wiley & Sons.