

Three Approaches to Generating Texts in Different Styles

Ehud Reiter¹ and Sandra Williams²

Abstract. Natural Language Generation (NLG) systems generate texts in English and other human languages from non-linguistic input data. Usually there are a large number of possible texts that can communicate the input data, and NLG systems must choose one of these. We argue that style can be used by NLG systems to choose between possible texts, and explore how this can be done by (1) explicit stylistic parameters, (2) imitating a genre style, and (3) imitating an individual's style.

1 Introduction

Natural Language Generation (NLG) systems are computer systems that automatically generate texts in English and other human languages, usually from non-linguistic input data. For example, NLG systems can generate textual weather forecasts from numerical weather prediction data [8, 22]; descriptions of museum artefacts from knowledge bases and databases that describe these artefacts [15]; information for medical patients based on their medical records [5, 6]; explanations of mathematical proofs based on the output of a theorem prover [10]; and so forth.

NLG systems essentially have to perform three kinds of processing [19]:

- *Document Planning:* Decide what information to communicate in the generated text. This is usually based on an analysis of the information needs of the reader of the text.
- *Microplanning:* Decide how the chosen content should be expressed linguistically; that is, what words and syntactic structures should be used, how information should be packaged up into sentences, and so forth.
- *Realisation:* Create an actual text based on the above decisions which is linguistically correct, and in particular conforms to the grammar of the target language.

In this paper, we focus on the second choice, deciding how to express information. In most cases there are dozens (if not thousands or even millions) of ways in which a piece of information can be expressed. Making such choices is one of the least understood aspects of NLG, and we believe that models of style (interpreted broadly) can be very useful tools in making such choices.

¹ Department of Computing Science, University of Aberdeen, Aberdeen AB24 3UE, UK. Email: e.reiter@abdn.ac.uk

² Department of Computing Science, The Open University, Milton Keynes MK7 6AA, UK. Email: s.h.williams@open.ac.uk

2 SkillSum

In order to make the following discussion concrete, we will use examples from SKILLSUM [25, 31], an NLG system which was developed by Aberdeen University and Cambridge Training and Development Ltd. SKILLSUM generates feedback reports for people who have just taken an on-line screening assessment of their basic literacy and numeracy skills. The input to the system is the responses to the questions on the assessment (an example assessment question is shown in Figure 1), plus some limited background information about the user (self-assessment of skills, how often he/she reads and writes, etc). The output is a short report (see example in Figure 2), which is intended to increase the user's knowledge of any problems that he or she has, and (if appropriate) encourage the user to enrol in a course to improve his or her basic skills.

SKILLSUM must perform the three tasks described above. Briefly (see architectural description in Figure 3):

- *Document planning:* SKILLSUM uses schemas [13] to choose content. That is, it chooses content based on a set of rules which were originally devised by analysing and 'reverse engineering' a set of human-written feedback reports, and which were then revised based on feedback from domain experts (basic skills tutors) and also from a series of pilot experiments with users [29].
- *Microplanning:* SKILLSUM uses a constraint-based approach to make expression choices. The SKILLSUM microplanner has a set of hard constraints and a preference function [30]. The hard constraints specify which choices and which combinations of choices are linguistically allowed. The preference function rates the choice sets; SKILLSUM chooses the highest scoring choice set allowed by the hard constraints. As discussed below, style seems especially useful in the context of the SKILLSUM preference function.
- *Realisation:* SKILLSUM includes two realisers, one of which operates on deep syntactic structures [11], and the other of which operates on template-like structures

To take a simple example of microplanning, suppose that SKILLSUM wants to tell a user that he got 20 questions right on the assessment, and that this is a good performance. A few of the many ways of saying this are:

- *You scored 20, which is very good.*
- *You scored 20. This is very good.*
- *You got 20 answers right! Excellent!*
- *Excellent, you got 20 answers right!*

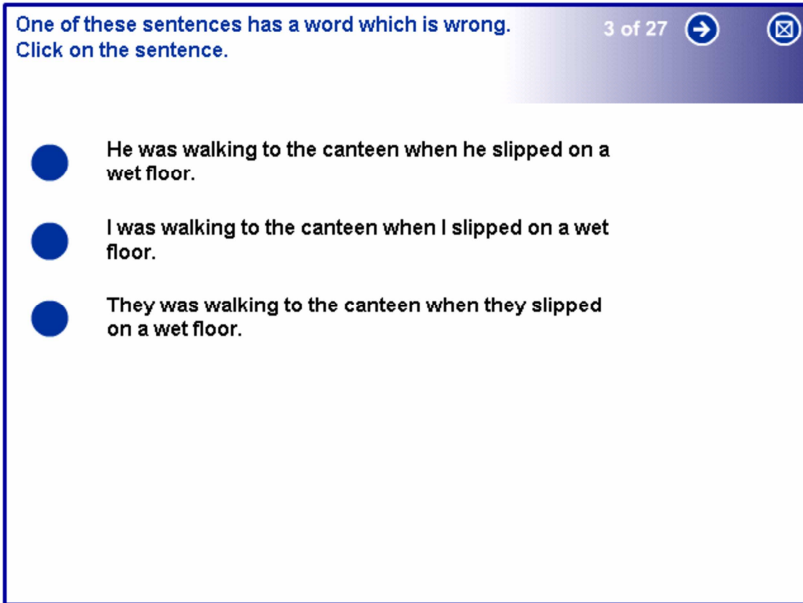


Figure 1. Example SkillSum Assessment Question

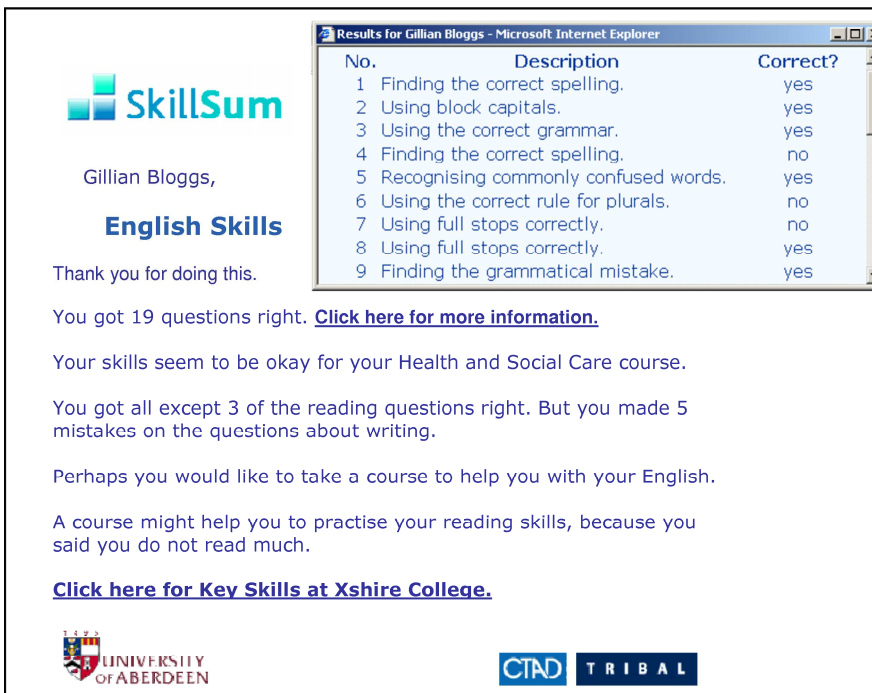


Figure 2. Example SkillSum Output Text

- 20 questions were answered correctly, this is a very good score.

The above examples illustrate some of the choices that are made in the microplanning process:

- *Lexical choice*: Which words should be used to communicate information? For example, should the first verb be *scored*, *got*, or *answered*?
- *Aggregation*: How should information be distributed among sentences? For example, should the above information be

communicated in one sentence or in two sentences?

- *Ordering*: What order should information be communicated in? In the above example, should the numerical score (20) or the qualitative assessment (e.g., *excellent*) come first?
- *Syntactic choice*: Which syntactic structures should be used? For example, should sentences be active voice (e.g., *You answered 20 questions . . .*) or passive voice (e.g., *20 questions were answered . . .*).
- *Punctuation*: For example, should full stops (“.”) or exclamation points (“!”) be used?

The above list is of course not exhaustive; for example it does not include deciding on referring expressions (e.g., *The big dog* vs. *Fido* vs. *it*), which is not very important in SKILLSUM, but is important in many other NLG applications. Decisions also of course have to be made in the other NLG stages (document planning and realisation), but we will focus on microplanning in this paper. We will also focus on how style affects words, syntax, and sentences, and ignore how style affects visual aspects of text such as layout [17].

3 Using Style to Make Microplanning Choices

One appealing way to make decisions about lexical choice, aggregation, and so forth is to appeal to psycholinguistic knowledge about the impact of texts on readers. For example, if an NLG system is trying to generate texts which are very easy to read (as was the case with SKILLSUM), it would be nice to base choices on psycholinguistic models of the impact of different words, sentence lengths, and so forth on reading speed and comprehension [9]. Similarly, if an NLG system is trying to generate texts which motivate or persuade people (such as STOP [20], which generated personalised smoking-cessation letters), it seems logical to base these choices on psycholinguistic models of how texts motivate and persuade people.

Unfortunately, our knowledge of psycholinguistics is imperfect, which makes this difficult to do. Also in practice context (such as how much sleep the reader had the previous night) can effect the psycholinguistic impact of different choices; and such contextual knowledge is usually not available to NLG systems. SKILLSUM in fact tried to base some of its choices on psycholinguistic models of readability, and while this worked to some degree, overall this strategy was less effective than we had hoped.

Another way to make choices is to look at frequency in large general English corpora, such as the British National Corpus (BNC) (<http://www.natcorp.ox.ac.uk/>) or one of the newspaper article corpora distributed by the Linguistic Data Consortium. Such corpora play a prominent role in much current research in Natural Language Processing.

For example, the average length of sentences in the BNC is 16 words. Hence we could base aggregation decisions on sentence length; for example we could say that two pieces of information should be aggregated and expressed in one sentence if and only if this aggregation brings average sentence length closer to 16 words/sentence. Of course aggregation decisions must consider other factors as well, such as semantic compatibility (for example, *John bought a radio and Sam bought a TV* is better than *John bought a radio and Sam bought an apple*).

A perhaps more basic problem is that rules based on a corpus which combines many types of texts intended for many audiences, such as the BNC, may not be appropriate for the context in which a specific NLG system is used. For example, because SKILLSUM users are likely to have below-average literacy skills, they should probably get shorter sentences than is the norm; indeed SKILLSUM sentences on average are only 10 words long.

Another problem with relying on a general corpus such as the BNC is that in many contexts there are strong conventions about choices, and these should be respected. For example, one version of SKILLSUM generated reports for teachers instead of for the people actually taking the test, and this version referred to test subjects as *learner*, because this is the standard term used by adult literacy tutors to refer to the people they are teaching. The perhaps more obvious word *student* is much more common in the BNC (it occurs 16 times more often than *learner*), and probably would be used in texts which used choice rules based on BNC frequency; but this would be a mistake, because teachers in this area have a strong convention of using the word *learner* instead of the word *student*.

Hence a better alternative is to try to imitate the choices made in a corpus of human-authored texts which are intended to be used in the same context as the texts we are trying to generate. This can be done in two ways: we can either collect a corpus of texts written by many authors which are representative of human-authored texts in this domain, or we can collect a corpus of texts from a single author, perhaps someone we believe is a particularly effective writer. In other words, we can try to imitate the **style** of texts in the genre as a whole, or the **style** of a particular individual author.

Yet another approach to making microplanning choices is to allow the reader to directly control these choices. In practice this seems most successful if choices are presented to the user as **stylistic** ones, such as level of formality.

These approaches are summarised in Table 1.

4 Style 1: Explicit Stylistic Control

Perhaps the most obvious solution to the choice problem is to directly ask users what choices they prefer in texts generated for them. After all, software which presents information graphically usually gives users many customisation options (colours, fonts, layout, etc), so why not similarly give users customisation options for linguistic presentations of information?

It is not feasible to ask users to directly specify microplanning choice rules, because there are too many of them; for example, SKILLSUM has hundreds of different constraints, and its preference functions contain dozens of components. Hence users are usually asked to specify a few high-level parameters which the NLG system then maps into the actual low-level microplanning choice rules. For example, rather than directly specify aggregation rules, a SKILLSUM user could specify a preferred average sentence length (either numerically or via a linguistic term such as *short*, *medium*, or *long*). This length could be used by the aggregation system as described above (Section 3). Similarly, rather than specify specific lexical choice rules for individual concepts, the user could specify whether he wants informal, moderately formal, or very formal

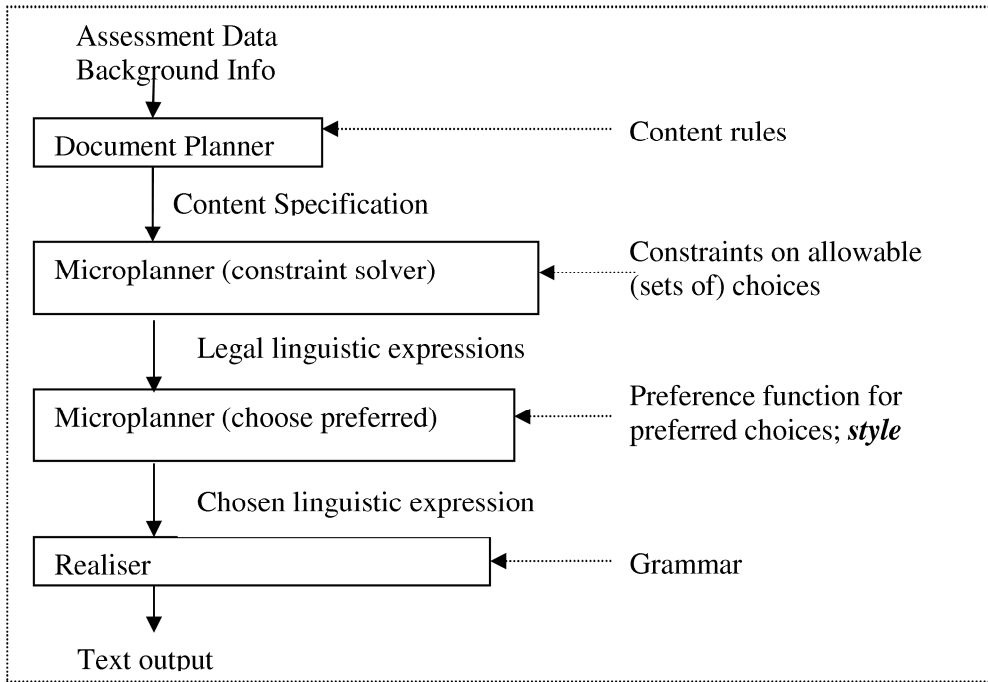


Figure 3. SkillSum architecture

Explicit Control	Allow the user to specify the choices that she prefers. Choices are usually presented as stylistic ones.
Conform to Genre	Imitate the choices made in a corpus of genre texts.
Imitate Individual	Imitate the choices made by an individual writer.

Table 1. Three ways of using style to control choices in NLG

language; whether he prefers common words with many meanings (such as *got*) or less common words with fewer meanings (such as *answered*); and so forth. These general preferences could then be examined by SKILLSUM’s detailed lexical choice rules. Such general preferences are usually perceived by users as *stylistic* preferences.

Although some of SKILLSUM’s internal choice rules did refer to general preferences such as frequency vs. number of meanings, SKILLSUM users were not allowed to directly control these. Instead, the SKILLSUM developers refined the rules and preferences based on feedback and suggestions from literacy teachers and students. In other words, users made change requests to a developer instead of directly controlling the system. This is not ideal, but it means we did not have to deal with the difficult problems of designing an appropriate user interface for soliciting preferences [24] and also ensuring that SKILLSUM was robust enough to generate appropriate texts for any preference settings, no matter how bizarre.

Other NLG projects have tried to explicitly allow users to specify high-level stylistic preferences. For example, WebbeDoc [7] allowed users to specify level of formality, amount of technical content and vocabulary, literacy level,

‘coolness’, and role (e.g., doctor or patient); the system then generated a text according to these stylistic parameters. WebbeDoc did this using a ‘Master Document’ which encoded a rich representation of information that could be communicated in a document, and ways this information could be expressed; WebbeDoc then selected appropriate pieces of the Master Document, based on the stylistic settings, and combined these into a generated text. WebbeDoc’s master document had to be carefully designed so that the above combination strategy did not result in incoherent texts. Perhaps the main long-term challenge in this approach is developing techniques, especially at the microplanning level, for automatically integrating master document segments into coherent texts. This may require changing some of these texts at the microplanning level, for example to ensure that appropriate referring expressions are used.

Another approach was taken by Paiva and Evans [16], who tried to base their controls on statistical analyses of texts. They analysed the surface linguistic features in a corpus of texts, and used factor analysis to cluster these on two dimensions. Their first dimension seemed to capture whether texts involved the reader or were distant from the reader; the second

focused on the type of references used (e.g., pronouns or full noun phrases). In other words, although the dimensions were produced by factor analysis, they seemed to capture some notion of what humans would call style. Paiva and Evans' analysis was inspired by Biber's analysis [4], although they used fewer dimensions (essentially because they were working with texts in a limited genre). Paiva and Evans then built an NLG system which could produce texts with user-specified values of their two dimensions. This system was based on a model of how individual microplanner choices affected these dimensions; this model was created by statistically analysing the ratings (in the two dimensions) of texts generated with random microplanner choices.

In short, while the WebbeDoc developers choose intuitively appealing stylistic dimensions and explicitly coded how these dimensions affected the generation process, Paiva and Evans used statistical techniques to derive both the dimensions and the rules that linked dimensions to actual generation decisions.

While both of the above systems are very interesting, it perhaps is worth pointing out that both have only been demonstrated to work on a small set of examples. It seems likely that there would be major engineering challenges in scaling either system up so that it could robustly generate large numbers of varied texts.

Another constraint-based NLG system, ICONOCLAST [18], enabled low-level style preferences such as paragraph length, sentence length, word length, technical terms, passive voice and graphical impact to be configured by manipulating sliders in a graphical user interface (www.itri.brighton.ac.uk/projects/iconoclast/walk/trial.html). A user's selections did not change the constraints directly, but instead the selections modified weights associated with violating soft constraints. These in turn were used to compute a cost function associated with each output text. Allowing soft constraints to be violated with varying costs offers one solution to the problem of introducing user preferences into constraint satisfaction problem solving. It is unfortunate that the ICONOCLAST style interface has not yet been evaluated with users, because there remains the problem of whether it is reasonable to expect users to understand how to choose sets of low-level style parameters. Intuitively, making such low-level language choices seems a difficult task. ICONOCLAST's Web interface also grouped low-level style preferences into higher-level style profiles such as, "broadsheet" and "tabloid", an approach that looks more promising in terms of usability. Indeed, determining suitable style profiles and evaluating their usability would be fruitful topics for future research.

An obvious source of existing style profiles would be the in-house style guidelines used by newspaper copy editors. Some newspapers publish their own style guidelines, e.g., the Guardian (www.guardian.co.uk/styleguide), however, these tend to have rather vague directives such as "vary sentence length" which would be hard to encode. What they might offer, though, are concrete lists of style features that these publications consider to be important. A comparative study of in-house style guidelines from different publishers would show whether publishers vary, and how.

5 Style 2: Conform to a Genre

Another approach to making choices is to imitate a corpus of human-written texts. As mentioned above, imitating a general corpus such as the BNC is problematical because it ignores constraints due to the domain, the genre, and the characteristics of the user population; these are very important in many NLG applications. However, we can try to imitate a corpus of texts written for the NLG system's application, domain, and users. In other words, we can analyse a corpus of human-written texts in the genre; learn the words, syntactic structures, and so forth that human writers use; and program our NLG system to imitate these choices.

This imitation can be done in a number of different ways. In particular, we can manually analyse the corpus and extract choice rules from it; we can automatically extract choice rules using statistical corpus analysis and statistical generation techniques; or we can use a combination of these techniques.

For example, when building SKILLSUM we collected a small corpus of 18 example human-written reports; these were written by two tutors (one of which specialised in literacy and one of which specialised in numeracy). We analysed this, mostly by hand (since the corpus was quite small), primarily to create hard constraints for SKILLSUM's microplanner. In other words, we tried to get SKILLSUM to generate appropriate genre texts by only allowing it to make choices which we observed in the corpus.

To take a concrete example, the corpus texts used the verbs *scored*, *answered*, and *got* (e.g., *you answered 20 questions correctly*); but they did not use the verbs *responded* (e.g., *you responded to 20 questions correctly*) or *aced* (e.g., *you aced 20 questions*). Hence a hard constraint on SKILLSUM is that it should not use *responded* or *aced*. In a sense, this suggests that SKILLSUM reports should be moderately formal; and if style was being explicitly specified as in Section 4, then this level of formality might be explicitly specified. But in the genre-corpus approach we don't specify such high-level stylistic parameters such as level of formality, instead we directly specify low-level choices such as which verbs can be used when communicating numerical performance on an assessment.

In a few cases we allowed SKILLSUM to deviate from the corpus; but this often proved ill-advised. For example, we programmed SKILLSUM to use *right* instead of *correct* or *correctly*, for example *you got 20 questions right* instead of *you got 20 questions correct*. We did this because *right* is much more common in the BNC, and hence we thought it would be easier to read. Although the tutors agreed that *right* could be used, when we asked 25 students enrolled in a literacy course about this choice, 23 (92%) preferred *correct* over *right*, and 24 (96%) preferred *correctly* over *right*. This suggests that allowing SKILLSUM to use a word which was not in the corpus, at least in this example, was a mistake.

Of course the SKILLSUM microplanner needs a preference function (to choose between allowable options) as well as hard constraints (to say which options should be considered). In theory preferences between choices can be specified by looking at frequencies, but this is more controversial. For example, in the SKILLSUM corpus *scored* is more common than *answered* or *got*, so *scored* should be preferred under a pure frequency-based metric. However, frequencies are not always

a good guide [22], because they may reflect the writing habits and preferences of a few individual corpus authors. In fact, *scored* was only used in reports written by one tutor, but it has the highest frequency because this tutor contributed the most texts to the SKILLSUM corpus. Hence in this case corpus frequency is really telling us about the linguistic preferences of the biggest contributor to the corpus; as we have no a priori reason to believe that this person is a better writer than the other corpus contributor, we need to interpret corpus frequency with caution.

In terms of methodology, SKILLSUM's rules were based on manual inspection of the corpus. Another possibility is to use machine learning techniques to automatically create rules or decision trees from a corpus; these can then be manually inspected by developers, who can modify the rules if necessary. This approach was used in SUMTIME-MOUSAM [22], which generated weather forecasts. SUMTIME-MOUSAM's microplanning rules (which focused on lexical choice, aggregation, and ellipsis) were based on careful analysis of a corpus of human-authored weather forecasts. Although most of these analyses were initially done using machine learning or statistical techniques, the rules suggested by the analyses were examined by developers and discussed with domain experts before they were added to the system [23]. This was especially important in cases where the corpus analysis showed that there was considerable variation in how different individuals made a choice. An evaluation with forecast users showed that the texts produced by SUMTIME-MOUSAM were very good, indeed in some cases they were perceived as being better than the human-written texts in the corpus.

Genre-specific microplanning rules can also be produced purely by machine learning and statistical analysis techniques, without having rules inspected by human developers or domain experts. This approach was used by Belz [1], who reimplemented some of SUMTIME-MOUSAM's functionality using a pure learning approach. An obvious advantage of this approach is that it is cheaper, since less human input is needed. Another advantage is that the rules do not have to be understandable by humans, as is the case with SUMTIME-MOUSAM's semi-automatic approach. However, a disadvantage is that developers, domain experts, and users cannot suggest that rules be modified based on their experience. An evaluation that compared Belz's system, SUMTIME-MOUSAM, and the human-written corpus texts [2] suggested that SUMTIME-MOUSAM's texts were on the whole better than Belz's texts, but Belz's texts were still quite good and in particular were sometimes better than the human-written corpus texts.

Perhaps the biggest problem we have faced in using machine learning techniques (whether semi-automatic or fully automatic) to learn microplanning choices in our NLG projects is obtaining a sufficiently large corpus. Although a few NLG systems such as SUMTIME-MOUSAM generate texts which are currently written by humans, it is more common for NLG systems to generate texts which are not currently manually written. In such cases it is not possible to get large corpora of naturally-occurring texts. In principle, one could analyse the microplanning choices made in related naturally-occurring texts, but this would require knowing which microplanning choices observed in the related texts could be applied to the NLG texts, and which could not.

In the SKILLSUM context, for example, domain experts (tu-

tors) do not currently write reports about the results of assessments, instead they orally discuss results with their students. We could in principle obtain a corpus of transcripts of discussions about assessments between tutors and students, and use learning and statistical techniques to analyse the choices made in the transcripts. But this is of limited utility unless we know which microplanning choices observed in the oral transcripts are also appropriate for written reports (lexical choice?), and which are not (aggregation?).

In other words, it would be much easier to use machine learning techniques to learn microplanning choices if we had a good understanding of which choices were stable across 'sub-styles' in a genre and which were not. Unfortunately, little currently seems to be known about this topic.

6 Style 3: Imitate a Person

A final style-related approach to making linguistic decisions is to imitate a person. As mentioned above, one problem with imitating a multi-author corpus is that different authors have different preferences between choices (in other words, different styles). Hence the frequencies in a corpus may reflect the choices of only a few authors who contributed the most texts rather than the best choice, as mentioned above. Also, choosing the most frequent choice every time may lead to inconsistencies which users dislike, essentially because this mixes the style of multiple authors [23].

An alternative is to try to imitate the linguistic choices made by a single person, perhaps someone who is known to be an effective writer in this genre. Imitating a single person increases consistency between choices, and also is likely to increase choice effectiveness if this person is an exceptionally good writer. However, a corpus from one individual is likely to be smaller and have worse linguistic coverage than a corpus with contributions from many people. Also, very good writers are likely to be very busy, which can make it difficult to discuss things directly with them.

SKILLSUM partially followed this approach when making decisions about content. More precisely, SKILLSUM generates two kinds of reports, literacy and numeracy, and the SKILLSUM corpus contains reports from two authors, one of whom is a literacy expert and the other of whom is a numeracy expert. When making some high-level decisions about the content of SKILLSUM's literacy reports, we tended to favour the choices made in the texts written by the literacy tutor; similar we focused on the numeracy tutor's choices when making choices about SKILLSUM's numeracy reports.

McKeown, Kukich, and Shaw [14] used this approach when building PlanDoc, an NLG system which produced summaries of the results of a simulation of changes to a telephone network. They interviewed a number of people to establish the general requirements of PlanDoc, but they asked a single very experienced domain expert to write all of the texts in their corpus. They do not give details of how they analysed and used the corpus, but it seems to have been a manual analysis rather than one based on learning or statistical techniques.

Another approach to individual style is to try to imitate the style of the *reader*, that is to generate texts in the style that the reader prefers when reading texts in this genre. Different people have different preferences. For example people who are very poor readers may do best with very short (5 word)

sentences, people who are moderately poor readers may prefer 10 word sentences, people with average skills may prefer 15-20 word sentences, etc. We could directly ask people about their preferences, as discussed in Section 4. However this approach is limited in that most people will probably only be willing to explicitly specify a small number of preferences.

Perhaps the most advanced work in this area is that of Walker and her colleagues [28]. They asked users to explicitly rate 600 texts generated by their NLG system with random microplanning choices. They employed learning techniques to determine which sets of microplanning choices produced texts preferred by each user, and from this created choice models for each user, which could be loaded into the microplanner. Their experiments suggested that users did indeed prefer texts generated using their personal choice models. Walker *et al* also commented that they believed reasonable individual choice models could be extracted from ratings of 120 texts, and getting this number of ratings is probably more realistic than getting 600 ratings from each user.

Walker *et al* did not really consider lexical choice, which is a shame because we know that there are substantial differences in the meanings that different individuals associate with words [21]. This has been reported in many contexts, including weather forecasts [22], descriptions of side effects of medication [3], and interpretation of surveys [26]. It was also an issue in SKILLSUM. For example, while developing SKILLSUM we asked 25 people enrolled in a literacy course to tell us what kind of mistake was in the sentence

I like apple's

72% said this was a *punctuation* mistake but 16% said this was a *grammar mistake* (the rest didn't think there was anything wrong with this sentence). Hence if we want to tell someone that he or she has problems with this kind of construct, we should probably refer to it as *grammar mistake* for the first group, and *punctuation mistake* for the second. Note that while this may sound like a small point, in fact some SKILLSUM users got quite annoyed when SKILLSUM told them they were bad at something which they thought they were good at. For example, if a SKILLSUM user made the above mistake and SKILLSUM told him he had problems with *punctuation*, the user might get annoyed if he interpreted *punctuation* to just mean commas and full stops (periods), since he had not made any mistakes with these.

Perhaps the key problem in doing this kind of tailoring is getting sufficient data about the individual; how do we actually find out how he or she uses words? If we only need data about a small number of lexical choices, that we could use an approach similar to Walker *et al*; but this is unlikely to be feasible if we need information about many different lexical choices.

An alternative approach might be to analyse a large corpus of texts that the user has written, on the assumption that the style used in texts the user writes is similar to the style preferred by the user in texts that he or she reads. Lin [12] looked at one aspect of this in her investigation of distributional similarities of verbs in a corpus of cookery writing to find alternatives for how different recipe authors expressed the same concept (e.g., “*roast* the meat in the oven” vs. “*cook* the meat in the oven”). To the best of our knowledge larger-scale investigations of larger sets of style choices have not yet been

tried; one concern is that many people (including most SKILLSUM users) do not write much, which would make it difficult to collect a reasonable corpus of their writings.

Data-scarcity becomes an even larger problem if we want to create models of individual linguistic preferences in specific genres. Ideally we would like not just a fixed set of linguistic preferences for a particular individual, but rather a mechanism for creating preference rules that express how text should be written for a particular individual in a specific genre. Again we are not aware of any existing research on this issue.

NLG might also employ research from the area of text categorisation and author identification, e.g.: Stamatatos *et al*. [27], that attempt to identify authorship using machine learning of word or character n-grams from a corpus. Being very speculative, if an NLG system could generate outputs with different combinations of microplanning parameters, perhaps an authorship identification system could be used to select the text which was most similar to the target author. Of course there are many issues that would need to be resolved before this could be done, not least of which is that existing author identification systems work with human-generated texts, not computer-generated texts.

7 Research Issues

As should be clear from the above, there are numerous research issues in this area that can be explored, for both technological reasons (building better NLG systems) and scientific reasons (enhancing our understanding of style). A few of these challenges are:

- *Explicit stylistic controls*: What stylistic controls make sense to human users, and how can these be ‘translated’ into the very detailed choices and preferences that control NLG microplanners?
- *Conform to a genre*: How are rules derived from a genre corpus most likely to differ from rules derived from a general corpus? In other words, how do genre texts actually differ from non-genre texts? Are there rules which are unlikely to vary, and hence could be derived from a general corpus?
- *Individual stylistic models*: How can we get good data about an individual’s language usage and preferences? What aspects of language usage are most likely to vary between individuals? How can we combine a (non-user-specific) genre language model with a (non-genre specific) individual language model?
- *What is the impact of style*: Generated texts can be evaluated in many different ways, including preference (e.g., do people like a text), readability (e.g., how long does it take to read a text), comprehension (e.g., how well do people understand a text), and task effectiveness (e.g., how well does a text help a user to do something). Which of these measures is most (and least) affected by adding stylistic information to an NLG system?

To conclude, we believe that style is an important aspect of generating effective and high-quality texts, and we are very pleased to see that an increasing number of NLG researchers are investigating style-related issues. We hope this research will lead to both better NLG systems, and also to a deeper scientific understanding of style in language.

Acknowledgements

We would like to thank our colleagues in Aberdeen and Milton Keynes, the anonymous reviewers, and the tutors we worked with in skillsum for their insightful comments and suggestions. This work was funded by PACCIT-LINK grant ESRC RES-328-25-0026.

REFERENCES

- [1] Anja Belz, 'Statistical generation: Three methods compared and evaluated', in *Proceedings of ENLG-2005*, pp. 15–23, (2005).
- [2] Anja Belz and Ehud Reiter, 'Comparing automatic and human evaluation of NLG systems', in *Proceedings of EACL-2006*, pp. 313–320, (2006).
- [3] Dianne Berry, Peter Knapp, and Theo Raynor, 'Is 15 per cent very common? informing people about the risks of medication side effects', *International Journal of Pharmacy Practice*, **10**, 145–151, (2002).
- [4] Douglas Biber, *Variation across speech and writing*, Cambridge University Press, 1988.
- [5] Bruce Buchanan, Johanna Moore, Diana Forsythe, Guiseppe Carenini, Stellan Ohlsson, and Gordon Banks, 'An interactive system for delivering individualized information to patients', *Artificial Intelligence in Medicine*, **7**, 117–154, (1995).
- [6] Alison Cawsey, Ray Jones, and Janne Pearson, 'The evaluation of a personalised health information system for patients with cancer', *User Modelling and User-Adapted Interaction*, **10**, 47–72, (2000).
- [7] Chrysanne DiMarco, Graeme Hirst, and Eduard H. Hovy, 'Generation by selection and repair as a method for adapting text for the individual reader', in *Proceedings of the Workshop on Flexible Hypertext, 8th ACM International Hypertext Conference*, (1997).
- [8] Eli Goldberg, Norbert Driedger, and Richard Kittredge, 'Using natural-language processing to produce weather forecasts', *IEEE Expert*, **9**(2), 45–53, (1994).
- [9] Trevor Harley, *The Psychology of Language*, Psychology Press, second edn., 2001.
- [10] Xiaorong Huang and Armin Fiedler, 'Proof verbalization as an application of NLG', in *Proceedings of IJCAI-1997*, volume 2, pp. 965–972, (1997).
- [11] Benoit Lavoie and Owen Rambow, 'A fast and portable realizer for text generation', in *Proceedings of the Fifth Conference on Applied Natural-Language Processing (ANLP-1997)*, pp. 265–268, (1997).
- [12] Jing Lin, 'Using distributional similarity to identify individual verb choice', in *Proceedings of the Fourth International Natural Language Generation Conference*, pp. 33–40, (2006).
- [13] Kathleen McKeown, *Text Generation*, Cambridge University Press, 1985.
- [14] Kathleen McKeown, Karen Kukich, and James Shaw, 'Practical issues in automatic document generation', in *Proceedings of ANLP-1994*, pp. 7–14, (1994).
- [15] Mick O'Donnell, Chris Mellish, Jon Oberlander, and Alistair Knott, 'ILEX: an architecture for a dynamic hypertext generation system', *Natural Language Engineering*, **7**, 225–250, (2001).
- [16] Daniel Paiva and Roger Evans, 'Empirically-based control of natural language generation', in *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pp. 58–65, Ann Arbor, Michigan, (June 2005). Association for Computational Linguistics.
- [17] Paul Piwek, Richard Power, Donia Scott, and Kees van Deemter, 'Generating multimedia presentations: From plain text to screenplay', in *Intelligent Multimodal Information Presentation*, Kluwer, (2005).
- [18] Richard Power, Donia Scott, and Nadjat Bouayad-Agha, 'Generating texts with style', in *Proceedings of the 4th International Conference on Intelligent Text Processing and Computational Linguistics (CICLing'03)*, pp. 444–452, (2003).
- [19] Ehud Reiter and Robert Dale, *Building Natural Language Generation Systems*, Cambridge University Press, 2000.
- [20] Ehud Reiter, Roma Robertson, and Liesl Osman, 'Lessons from a failure: Generating tailored smoking cessation letters', *Artificial Intelligence*, **144**, 41–58, (2003).
- [21] Ehud Reiter and Somayaajulu Sripada, 'Human variation and lexical choice', *Computational Linguistics*, **28**, 545–553, (2002).
- [22] Ehud Reiter, Somayaajulu Sripada, Jim Hunter, and Jin Yu, 'Choosing words in computer-generated weather forecasts', *Artificial Intelligence*, **167**, 137–169, (2005).
- [23] Ehud Reiter, Somayaajulu Sripada, and Roma Robertson, 'Acquiring correct knowledge for natural language generation', *Journal of Artificial Intelligence Research*, **18**, 491–516, (2003).
- [24] Ehud Reiter, Somayaajulu Sripada, and Sandra Williams, 'Acquiring and using limited user models in NLG', in *Proceedings of the 2003 European Workshop on Natural Language Generation*, (2003). Forthcoming.
- [25] Ehud Reiter, Sandra Williams, and Leslie Crichton, 'Generating feedback reports for adults taking basic skills tests', in *Applications and Innovations in Intelligent Systems XIII: Proceedings of AI-2005*, pp. 50–63. Springer, (2005).
- [26] Michael Schober, Frederick Conrad, and Scott Fricker, 'Misunderstanding standardized language in research interviews', *Applied Cognitive Psychology*, **18**, 169188, (2004).
- [27] Efstathios Stamatatos, Nikos Fakotakis, and George Kokkinakis, 'Automatic text categorization in terms of genre and author', *Computational Linguistics*, **26**, 471–495, (2000).
- [28] Marilyn Walker, Amanda Stent, François Mairesse, and Rashi Prasad, 'Individual and domain adaptation in sentence planning for dialogue', *Journal of Artificial Intelligence Research*, **30**, 413–456, (2007).
- [29] Sandra Williams and Ehud Reiter, 'Deriving content selection rules from a corpus of non-naturally occurring documents for a novel NLG application', in *Proceedings of Corpus Linguistics workshop on using Corpora for NLG*, (2005).
- [30] Sandra Williams and Ehud Reiter, 'Generating readable texts for readers with low basic skills', in *Proceedings of ENLG-2005*, pp. 140–147, (2005).
- [31] Sandra Williams and Ehud Reiter, 'Generating basic skills reports for low-skilled readers', *Natural Language Engineering*, (in press).