Chapter 4
Generating Texts in Different Styles

Ehud Reiter[1] and Sandra Williams[2]

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. This decision can partially be based on style (interpreted broadly). We explore three mechanisms for incorporating style into NLG choice-making: (1) explicit stylistic parameters, (2) imitating a genre style, and (3) imitating an individual's style.

4.1 Introduction

Natural Language Generation (NLG) systems are computer systems that automatically generate texts in English and other human languages, using advanced techniques from artificial intelligence and/or computational linguistics. In this chapter we focus on NLG systems whose goal is to present, summarise, or explain non-linguistic input data to users (the generation of poetry and other fictional material is discussed by Goguen and Harrell in Chapter 11). Example applications include generating textual weather forecasts from numerical weather prediction data [10, 25]; producing descriptions of museum artefacts from knowledge bases and databases that describe these artefacts [19]; providing information for medical patients based on their medical records [7, 8]; and creating explanations of mathematical proofs based on the output of a theorem prover [13].

The challenge of NLG is making choices about the content and language of the generated text. Considering content, for example, how much detail should a weather forecast text go into (e.g., *Sunday will be wet* or *There will be heavy rain on Sunday afternoon*); and considering language, should a

Department of Computing Science, University of Aberdeen, UK e.reiter@abdn.ac.uk · Department of Computing Science, The Open University, UK s.h.williams@open.ac.uk

weather forecast be in normal English or abbreviated 'weatherese' (e.g., *There will heavy rain on Sunday afternoon* or *Heavy rain on Sunday afternoon*). Sometimes these choices can be explicitly based on usefulness criteria; for example content choices can be motivated on the basis of user needs (e.g., farmers typically need more detailed information than lorry drivers), and linguistic choices can be motivated by readability considerations (e.g., short common words are usually read more quickly than long unusual words). But usually explicit usefulness criteria can only motivate some of the choices an NLG system must make; we need other criteria for making the remaining choices.

In this chapter we suggest that many NLG choices can be made on the basis of *style*, by which we mean the expectations or preferences of a particular user and/or document genre. There are many similarities between our analysis and and Dannenburg's discussion of style in music in Chapter 3; many of the detailed decisions in both textual and musical composition reflect the idiosyncrasies of an individual person and/or genre.

We focus here on how style can influence linguistic choices, because these are better understood than content choices. In particular, we present three techniques that have been used in NLG to adjust generated texts according to individual or genre characteristics: giving users explicit control over features of the generated text (so they can make it conform to their preferences); generating texts which are similar to a corpus of human-written texts in a particular genre; and generating texts which imitate the writing style of a specific individual. Our emphasis is on choices that affect words, syntax, and sentences; we do not discuss visual aspects of text such as layout [6].

A major problem with all of the above techniques is acquiring the necessary knowledge about the typical choices of a particular user, genre, or individual writer. Some information about these choices can be acquired using the computational stylistics techniques developed to analyse texts (for example, the author identification techniques discussed by Argamon and Koppel in Chapter 5), but unfortunately feature sets which are adequate for identifying an author are usually inadequate for reproducing his or her style. For example, we may be able to identify an author on the basis of frequency of function words, but an NLG choice-making system needs much more information than just function word frequency.

4.2 SkillSum

In order to make the following discussion concrete, we will use examples from SkillSum [32], an NLG system which was developed by Aberdeen University and Cambridge Training and Development Ltd (now part of Tribal Group). SkillSum generates feedback reports for people who have just taken an on-line screening assessment of their basic literacy and numeracy skills.

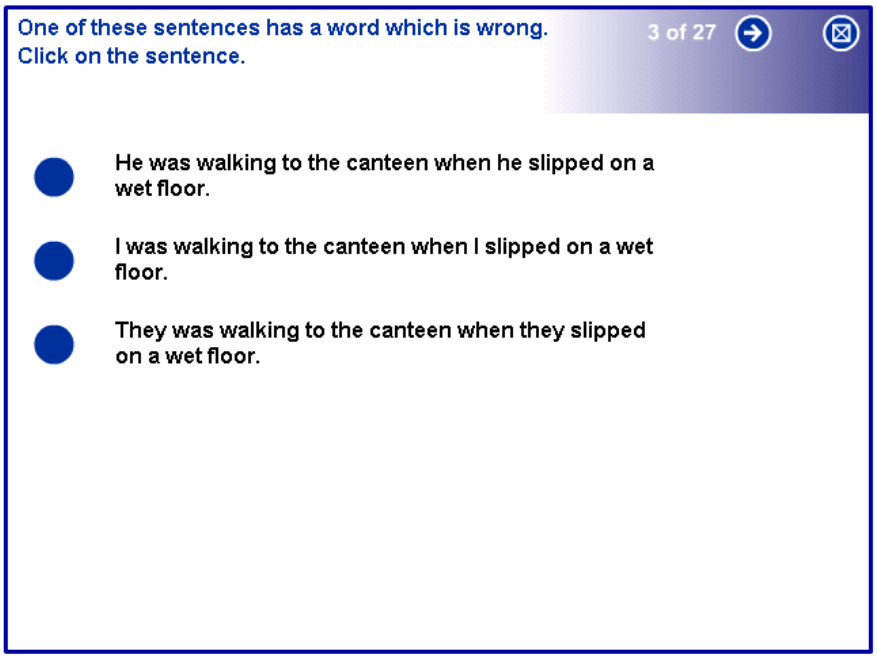

Fig. 4.1 Example SkillSum Assessment Question

The input to the system is the responses to the questions in the assessment (an example assessment question is shown in Figure 4.1), 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 4.2), 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, like most NLG systems, must perform three basic tasks [22]: decide on content and structure (*document planning*), decide how information should be expressed linguistically (*microplanning*), and generate an actual text based on the above decisions which is linguistically correct (*realisation*). Briefly (see architectural description in Figure 4.3):

- *Document planning:* SkillSum uses schemas [17] to choose content. That is, it chooses content by applying 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 in accordance with feedback from domain experts (basic-skills tutors) and also from a series of pilot experiments with users [30].
- *Microplanning:* SkillSum uses a constraint-based approach to make expression choices. The SkillSum microplanner has a set of hard constraints and a preference function [31]. The hard constraints specify which choices

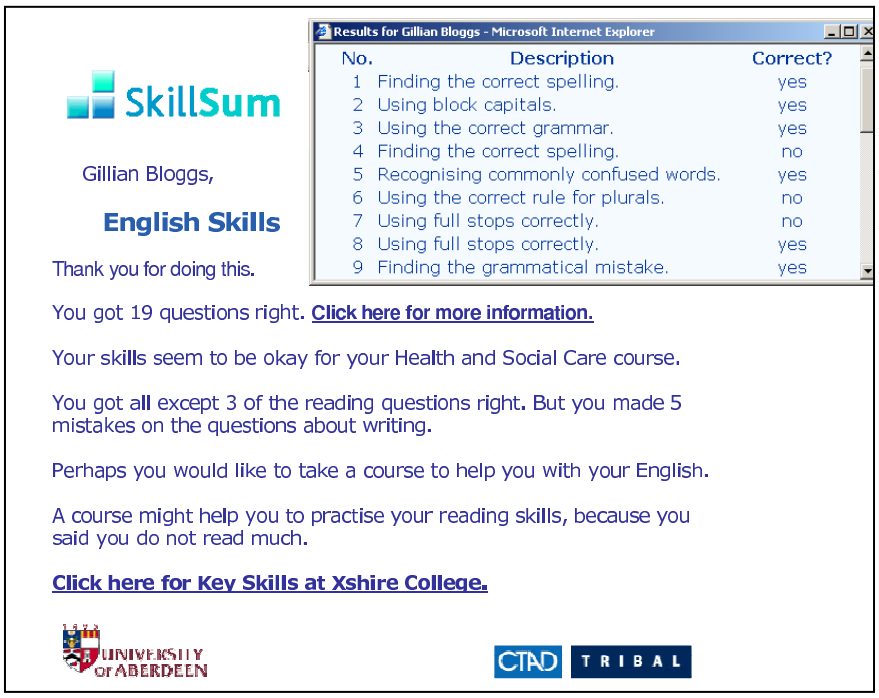

Fig. 4.2 Example SkillSum Output Text

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 one operates on deep syntactic structures [15] and the other on template-like structures

In this chapter we focus on microplanning, as this is where many NLG systems make linguistic choices. To take a simple example of microplanning, suppose that the SkillSum document planner has decided 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!*
- *20 questions were answered correctly; this is a very good score.*

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

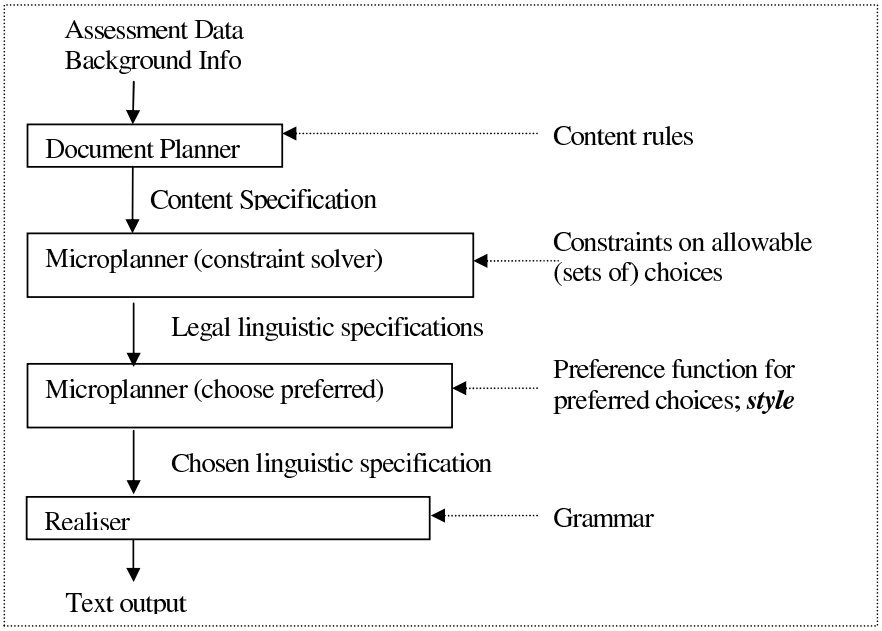

Fig. 4.3 SkillSum architecture

- *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 have 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 it is in many other NLG applications. Also it does not include content choices made by the document planner, such as the decision to give the user a numerical score.

4.3 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 [11]. Similarly, if an NLG system is trying to generate texts which motivate or persuade people as did STOP [23], 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 not detailed enough to serve this purpose. Also in practice context (such as how much sleep the reader had the previous night) can affect 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 normal; 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 and 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.

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 4.1 Three ways of using style to control choices in NLG

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 4.1.

4.4 Style 1: Explicit Stylistic Control

Perhaps the most obvious solution to the choice problem is to allow users to directly control some stylistic parameters when they run the generator. After all, software that 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 takes into consideration when making the actual low-level microplanning choices.

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 4.3). Similarly, rather than specify specific lexical choice rules for individual concepts, the user

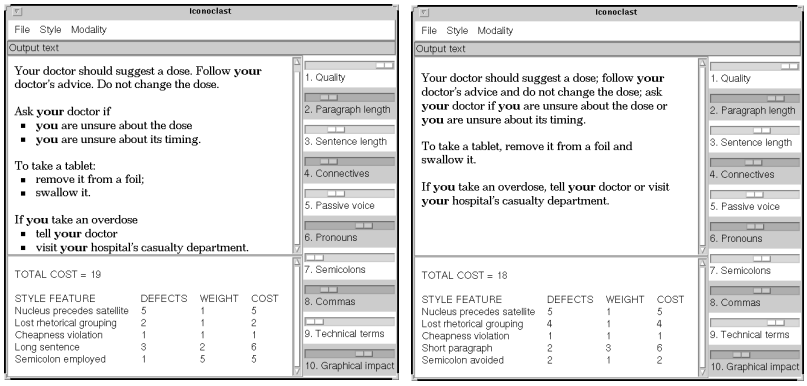

Fig. 4.4 User control examples from Iconoclast (from www.itri.brighton.ac.uk/projects/iconoclast/walk/trial.html)

could specify whether he wants informal, moderately formal, or very formal 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.

This approach was used in the ICONOCLAST system [21]. ICONOCLAST users could specify a number of high-level parameters, such as paragraph length, sentence length, and pronominalisation, by manipulating slider bars in a graphical user interface; see Figure 4.4 for an example. ICONOCLAST varied the style of the output by searching for solutions to a constraint satisfaction problem formulated from lower-level parameters and solutions were ordered according to a cost function derived from a user's slider bar selections. ICONOCLAST also grouped style preferences into higher-level style profiles such as 'broadsheet' and 'tabloid', which users could select.

A key question with this type of interface is the level of detail at which stylistic information should be specified by users; is it reasonable to expect users to specify values for parameters such as sentence length, or is it better to ask them to select from higher-level concepts such as 'tabloid', or indeed for more detailed information about specific microplanning choices? This is a human-computer interaction question, which can only be answered by detailed user studies (ideally studies which involve different types of users). Unfortunately, to the best of our knowledge no such studies have yet been done.

In the case of SkillSum, although some of its internal choice rules did refer to general preferences such as frequency vs. number of meanings, SkillSum users were not allowed to directly specify these. Instead, the developers refined the rules and preferences in accordance with feedback and suggestions from literacy teachers and students. In other words, users requested changes

from a developer instead of directly controlling the system [27]. This is not ideal, but it meant we did not have to deal with user-interface issues. Also it made SkillSum easier to debug, as we did not need to ensure that it would generate appropriate texts for any values of the stylistic preferences, no matter how bizarre.

From an algorithmic perspective, both iconoclast and SkillSum incorporated user-specified stylistic preferences into a preference function which was used by the microplanner to make specific decisions. Another approach was used by WebbeDoc [9] which, like iconoclast, allowed users to specify values of stylistic parameters using a GUI. WebbeDoc used the user-specified preferences to select and adapt material from a manually-authored Master Document, which specified different ways in which information could be expressed linguistically.

Yet another approach was taken by Paiva and Evans [20], who based their stylistic features on factor analysis of texts [5], instead of on features which were intuitively meaningful to users. This analysis produced two dimensions; the first seemed to capture whether texts involved the reader or were distant from the reader, and the second focused on the type of references used (e.g., pronouns or full noun phrases). 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.

It is unfortunate that iconoclast, WebbeDoc and Paiva and Evan's system were not properly evaluated by users. Without such an evaluation, it is difficult to know which approach is most promising for producing appropriate texts, and which stylistic features users are willing to specify.

4.5 Style 2: Conform to a Genre

Another approach to making choices is to imitate a corpus written for a particular genre. As mentioned above, imitating a very general corpus such as the BNC is problematical because it contains documents from a variety of different domains and genres that were written for a variety of different audiences. If it is important that generated texts conform to a genre (which is often the case), an alternative is to analyse a corpus of human-written texts in the target genre; learn the words, syntactic structures, and so forth that human writers select in the genre; and program the nlg system to imitate those choices.

4.5.1 Genre Modelling with Manual Corpus Analysis

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. In some cases it may even be possible to extract rules directly from published style guidelines, such as those used by newspapers (e.g., www.guardian.co.uk/styleguide). Most of these guidelines are probably too vague to be of much use to NLG, but there are some exceptions which are more detailed, such as AECMA Simplified English [1].

For example, when building SkillSum we collected a small corpus of 18 human-written feedback reports; these were written by two tutors (one of whom specialised in literacy and one in numeracy). We analysed it, 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 4.4, 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.

Formal text
In early April, a shantytown - named Winnie Mandela City - was erected by several students on Beinecke Plaza, so that Yale University would divest from companies doing business in South Africa.
Later, at 5:30am on April 14, the shantytown was destroyed by officials; also, at that time, the police arrested 76 students. Several local politicians and faculty members expressed criticism of Yale's action. Finally, Yale gave the students permission to reassemble the shantytown there and, concurrently, the University announced that a commission would go to South Africa in July to investigate the system of apartheid.

Informal text
Students put a shantytown, Winnie Mandela City, up on Beinecke Plaza in early April. The students wanted Yale University to pull their money out of companies doing business in South Africa.
Officials tore it down at 5:30am on April 14, and the police arrested 76 students. Several local politicians and faculty members criticised the action. Later, Yale allowed the students to put it up there again. The University said that a commission would go to South Africa in July to study the system of apartheid.

Fig. 4.5 Formal and informal texts from Pauline (Hovy, 1990)

However, frequencies are not always a good guide [25], 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.

Perhaps the most ambitious attempt to create genre-tailoring rules based on manual corpus analysis was Pauline [12]. Figure 4.5 shows two texts produced by Pauline, one formal and the other informal. These differ in numerous ways, including lexical choice, syntactic choice, use of pronominal referring expressions and aggregation. Although Pauline's output is very impressive, it relied on extensive hand-crafted knowledge bases which only covered a few scenarios. It would be difficult to build a system based on Pauline's approach which could robustly handle any plausible input data set in a realistic application domain.

4.5.2 Genre Modelling with Machine Learning and Statistics

In terms of methodology, SkillSum and Pauline were based on manual inspection of a 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 [25], which generated weather forecasts. SumTime'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 [26]. This was especially important in cases where the corpus analysis showed that there was considerable variation in the different choices that individuals made; feedback from domain experts in such cases helped us decide which choice was most likely to be appropriate for the largest number of readers. An evaluation with forecast users showed that the texts produced by SumTime 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 [2], who reimplemented some of SumTime'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's semi-automatic approach. However, a disadvantage is that developers, domain experts, and users cannot suggest that rules be modified in accordance with their experience. An evaluation that compared Belz's system, SumTime, and the human-written corpus texts [3] suggested that SumTime'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 generate texts which were previously 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 (tutors) 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 'substyles' in a genre and which were not. Unfortunately, little currently seems to be known about this topic.

4.6 Style 3: Imitate a Person

A final style-related approach to making linguistic decisions is to imitate either an individual *author*'s style, or to imitate the style of the texts that an individual *reader* prefers.

4.6.1 Imitate an Author

As mentioned above, one problem with imitating a multi-author corpus is that different authors have different preferences (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 consensus choice, as mentioned above. Also, if the system selects the most frequent choice every time, this may lead to inconsistencies which users dislike, essentially because it will mix the styles of multiple authors [26].

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 author increases consistency between choices, and also is likely to increase the quality of the generated texts 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 style issues 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, of whom one is a literacy expert and the other one 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; similarly we used the numeracy tutor's preferences when making choices about SkillSum's numeracy reports.

McKeown, Kukich. and Shaw [18] 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

Preferred text according to User A choice model:
Chanpen Thai is a Thai restaurant, with good food quality. It has good service. Its price is 24 dollars. It has the best overall quality among the selected restaurants.

Preferred text according to User B choice model:
Chanpen Thai has the best overall quality among the selected restaurants since it is a Thai restaurant, with good service, its price is 24 dollars, and it has good food quality.

Fig. 4.6 Texts generated from different personal preference models (Walker et al, 2007).

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.

4.6.2 Imitate the Style of the Texts that a Reader Prefers

Another approach to individual style is to try to imitate the style of the texts that a *reader* likes to read – that is, to ask the reader to choose between texts written in different styles, and generate the style that the reader prefers. Different people have different preferences. For example, some people may prefer texts with many pronouns, while others prefer texts with more explicit references (perhaps because of differences in their working memories [14]). We could directly ask people about their preferences, as discussed in Section 4.4. However this approach is limited in that many people will probably not be aware of the linguistic implications of their style preferences, nor may they be willing to explicitly specify more than a small number of preferences.

Perhaps the most advanced work in this area is that of Walker and her colleagues [29]. 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 a preference model for each user, which predicted how users would rank texts produced with different microplanner choices; Figure 4.6 shows examples of preferred texts for different users. Their experiments suggested that users did indeed prefer texts generated using their personal preference 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 preferences, which is a shame because we know that there are substantial differences in the meanings that different individuals associate with words [24]. This has been reported in

many contexts, including weather forecasts [25], descriptions of side effects of medication [4], and interpretation of surveys [28]. 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 occurred in the sentence *I like apple's* (instead of *I like apples*). 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).

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, then 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 [16] 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 more comprehensive 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.

4.7 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?
- *Conformity 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?
- *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, and an increasing number of computational stylistics researchers are interested in NLG. 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. We also thank the attendees of the AISB 2008 Symposium on 'Style in Text: Creative Generation and Identification of Authorship' (where we presented an earlier version of this paper) for their help and suggestions. This work was funded by PACCIT-LINK grant ESRC RES-328-25-0026.

References

1. AECMA: A guide for the preparation of aircraft maintenance documentation in the international aerospace maintenance language (1986). Available from BDC Publishing Services, Slack Lane, Derby, UK
2. Belz, A.: Statistical generation: Three methods compared and evaluated. In: Proceedings of ENLG-2005, pp. 15–23 (2005)
3. Belz, A., Reiter, E.: Comparing automatic and human evaluation of NLG systems. In: Proceedings of EACL-2006, pp. 313–320 (2006)
4. Berry, D., Knapp, P., Raynor, T.: Is 15 per cent very common? informing people about the risks of medication side effects. International Journal of Pharmacy Practice **10**, 145–151 (2002)
5. Biber, D.: Variation across speech and writing. Cambridge University Press (1988)

6. Bouayad-Agha, N., Scott, D., Power, R.: The influence of layout on the interpretation of referring expressions. In: L. Degand, Y. Bestgen, W. Spooren, L. van Waes (eds.) Multidisciplinary Approaches to Discourse, pp. 133–141. Stichting Neerlandistiek VU Amsterdam and Nodus Publikationen Munster (2001)
7. Buchanan, B., Moore, J., Forsythe, D., Carenini, G., Ohlsson, S., Banks, G.: An interactive system for delivering individualized information to patients. Artificial Intelligence in Medicine **7**, 117–154 (1995)
8. Cawsey, A., Jones, R., Pearson, J.: The evaluation of a personalised health information system for patients with cancer. User Modelling and User-Adapted Interaction **10**, 47–72 (2000)
9. DiMarco, C., Hirst, G., Hovy, E.H.: 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)
10. Goldberg, E., Driedger, N., Kittredge, R.: Using natural-language processing to produce weather forecasts. IEEE Expert **9**(2), 45–53 (1994)
11. Harley, T.: The Psychology of Language, second edn. Psychology Press (2001)
12. Hovy, E.: Pragmatics and natural language generation. Artificial Intelligence **43**, 153–198 (1990)
13. Huang, X., Fiedler, A.: Proof verbalization as an application of NLG. In: Proceedings of IJCAI-1997, vol. 2, pp. 965–972 (1997)
14. Just, M., Carpenter, P.: A capacity theory of comprehension: Individual differences in working memory. Psychology Review **99**, 122–149 (1992)
15. Lavoie, B., Rambow, O.: 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)
16. Lin, J.: Using distributional similarity to identify individual verb choice. In: Proceedings of the Fourth International Natural Language Generation Conference,, pp. 33–40 (2006)
17. McKeown, K.: Text Generation. Cambridge University Press (1985)
18. McKeown, K., Kukich, K., Shaw, J.: Practical issues in automatic document generation. In: Proceedings of ANLP-1994, pp. 7–14 (1994)
19. O'Donnell, M., Mellish, C., Oberlander, J., Knott, A.: ILEX: an architecture for a dynamic hypertext generation system. Natural Language Engineering **7**, 225–250 (2001)
20. Paiva, D., Evans, R.: 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. Association for Computational Linguistics, Ann Arbor, Michigan (2005)
21. Power, R., Scott, D., Bouayad-Agha, N.: Generating texts with style. In: Proceedings of the 4th International Conference on Intelligent Text Processing and Computational Linguistics (CICLing'03), pp. 444–452 (2003)
22. Reiter, E., Dale, R.: Building Natural Language Generation Systems. Cambridge University Press (2000)
23. Reiter, E., Robertson, R., Osman, L.: Lessons from a failure: Generating tailored smoking cessation letters. Artificial Intelligence **144**, 41–58 (2003)
24. Reiter, E., Sripada, S.: Human variation and lexical choice. Computational Linguistics **28**, 545–553 (2002)
25. Reiter, E., Sripada, S., Hunter, J., Yu, J.: Choosing words in computer-generated weather forecasts. Artificial Intelligence **167**, 137–169 (2005)
26. Reiter, E., Sripada, S., Robertson, R.: Acquiring correct knowledge for natural language generation. Journal of Artificial Intelligence Research **18**, 491–516 (2003)
27. Reiter, E., Sripada, S., Williams, S.: Acquiring and using limited user models in NLG. In: Proceedings of the 2003 European Workshop on Natural Language Generation (2003)
28. Schober, M., Conrad, F., Fricker, S.: Misunderstanding standardized language in research interviews. Applied Cognitive Psychology **18**, 169–188 (2004)

29. Walker, M., Stent, A., Mairesse, F., Prasad, R.: Individual and domain adaptation in sentence planning for dialogue. Journal of Artificial Intelligence Research **30**, 413–456 (2007)
30. Williams, S., Reiter, E.: 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)
31. Williams, S., Reiter, E.: Generating readable texts for readers with low basic skills. In: Proceedings of ENLG-2005, pp. 140–147 (2005)
32. Williams, S., Reiter, E.: Generating basic skills reports for low-skilled readers. Natural Language Engineering **14**, 495–535 (2008)