

Avoiding Repetition in Generated Text

Mary Ellen Foster

Informatik VI: Robotics and Embedded Systems
Technische Universität München
Boltzmannstr. 3, 85748 Garching, Germany
foster@in.tum.de

Michael White

Department of Linguistics
The Ohio State University
Columbus, OH 43210 USA
mwhite@ling.osu.edu

Abstract

We investigate two methods for enhancing variation in the output of a stochastic surface realiser: choosing from among the highest-scoring realisation candidates instead of taking the single highest-scoring result (ϵ -best sampling), and penalising the words from earlier sentences in a discourse when generating later ones (*anti-repetition scoring*). In a human evaluation study, subjects were asked to compare texts generated with and without the variation enhancements. Strikingly, subjects judged the texts generated using these two methods to be better written and less repetitive than the texts generated with optimal n -gram scoring; at the same time, no significant difference in understandability was found between the two versions. In analysing the two methods, we show that the simpler ϵ -best sampling method is considerably more prone to introducing dispreferred variants into the output, indicating that best results can be obtained using anti-repetition scoring with strict or no ϵ -best sampling.

1 Introduction

A classic rule of writing, found in many style guides, is to avoid repetition in order to keep text interesting and make it more lively. When designing systems to automatically generate text, it is often taken for granted that this stylistic goal should be met as well: for example, van Deemter et al. (2005) incorporated random choice into a language generation system “to maximise the *variety* of sentences produced” (emphasis original).

Repetitiveness may take several forms: using the same words or syntactic structures, repeatedly giving the same facts, or even repeating entire turns (for example, error-handling turns in dialogue systems).

At the level of word choice and phrasing, recent advances in stochastic text generation have made it possible to implement corpus-based approaches to varying output. However, as Stone et al. (2004) note, there is an inherent conflict between producing output that is optimally similar to the corpus and incorporating variability: varying output requires choosing less frequent options, which inevitably reduces scores on corpus similarity measures. To the extent that corpus-based measures (such as n -gram scores) are used to avoid overgeneration and select preferred paraphrases, it is not obvious how to enhance variation without reducing output quality.

With this question in mind, we investigate in this paper the impact of two different methods for enhancing variation in the output generated by the COMIC multimodal dialogue system.¹ Both methods take advantage of the periphrastic ability of the OpenCCG surface realiser (White, 2006a). In the usual OpenCCG realisation process, when a logical form is transformed into output text, n -gram models are used to steer the realiser towards the single highest-scoring option for the sentence. This process tends to select the same syntactic structure for every sentence describing the same feature: for example, in the COMIC domain (describing and comparing bathroom tiles), the structure *The colours are [colours]* would be used every time the colours of a tile design are to be presented, even though alternative paraphrases are available.

The first (and simplest) means of avoiding such repetition using OpenCCG, ϵ -best sampling, is to perform n -best realisation and then to select randomly from among those options whose score is within a threshold ϵ of the top score. The second

¹<http://www.hcrc.ed.ac.uk/comic/>

means of adding variation, *anti-repetition scoring*, is to store the words from recently generated sentences and to penalise a proposed realisation based on the number of words that it shares with these sentences. OpenCCG provides a built-in facility for implementing such anti-repetition scorers and integrating them with the normal n -gram-based scoring algorithm (White, 2005).

To verify that it can be beneficial for a natural language generation system to strive to avoid repetition, we first conducted a human evaluation study in which subjects were asked to compare texts generated with and without the two variation-enhancing methods. Strikingly, subjects judged the versions generated using ϵ -best sampling and anti-repetition scoring to be both better written and less repetitive than the versions generated with optimal n -gram scoring. To our knowledge, this study is the first to show a clear benefit for enhancing variation; while other recent studies (e.g., Stent et al., 2005; Belz and Reiter, 2006) have shown that automatic evaluation metrics do not always correlate well with human judgments of high quality generated texts with periphrastic variations, these studies examined sentences out of context, and thus could not take into account the benefit of avoiding repetition as a discourse progresses.

Following the human evaluation study, we varied the main parameters used in ϵ -best sampling and anti-repetition scoring and analysed the resulting impact on the amount of periphrastic variation and the number of dispreferred paraphrases in the generated outputs. The analysis revealed that the simpler ϵ -best sampling method is considerably more prone to introducing dispreferred variants into the output. It also showed that essentially the same amount of variation can be achieved using anti-repetition scoring on its own, or just with strict ϵ -best sampling, as using both methods together. This suggests a way of resolving the conflict between enhancing variation and maximising corpus similarity.

The rest of this paper is structured as follows. In Section 2, we describe previous work on generating paraphrases. In Section 3, we next summarise the realisation process of the OpenCCG surface realiser, concentrating on its use of n -gram models in the generation process and its support for disjunctive logical forms. In Section 4, we then give details of how the two anti-repetition methods were integrated into this realisation algorithm. Section 5 next presents the result of the human evaluation study.

In Section 6, we then explore the impact of the two anti-repetition methods on the variability and quality of the generated text, using a range of parameter settings. In Section 7, we discuss the results of both studies and compare them with related work. Finally, in Section 8, we give some conclusions and outline possible extensions to this work.

2 Previous Work

The acquisition and generation of paraphrases has been studied for some time (cf. Iordanskaja et al., 1991; Langkilde and Knight, 1998; Barzilay and McKeown, 2001; Barzilay and Lee, 2003; Pang et al., 2003). Much recent work in this area has focussed on the automated acquisition of paraphrases from corpora, along with the use of the resulting paraphrases in language-processing areas such as information extraction and retrieval, question-answering, and machine translation.

The main technique that has been used for adding variation to stochastically-generated output is to modify the system so that it does not always choose the same option in a given situation, normally by modifying either the weights or the selection strategy. When selecting a combination of speech and body-language output for an animated character based on a corpus of recorded behaviour, for example, Stone et al. (2004) introduced variation by perturbing the scores slightly to choose from among low-cost utterances. The outputs from the system with perturbed weights scored nearly as high on an automated evaluation as those from the optimised system, and also made use of a wider range of corpus data. Belz and Reiter's (2006) "greedy roulette" pCRU text-generation system selected among generation rules weighted by their corpus probabilities, while Foster and Oberlander (2006) used a similar technique to select facial displays for an animated talking head. Both of these systems scored higher on a human evaluation than at least one competing system that always chose the single highest-scoring option; see Section 7 for further discussion.

The CRAG-2 system (Isard et al., 2006) generates dialogues between pairs of agents who are linguistically distinguishable but able to align with each other. It uses the OpenCCG surface realiser to select appropriate paraphrases for the desired personality of the simulated character and the stage of the dialogue, integrating cache models built from the preceding discourse with the primary n -gram models to attain lexico-syntactic alignment. The

method of anti-repetition scoring described in this paper is similar, but the goal is opposite: instead of increasing alignment with an interlocutor, here we modify the n -gram scores to avoid alignment with the system’s own previous utterances.

3 Surface Realisation with OpenCCG

The studies described in this paper use the OpenCCG open source surface realiser (White, 2006a,b), which is based on Steedman’s (2000) Combinatory Categorical Grammar (CCG). A distinguishing feature of OpenCCG is that it uses a hybrid symbolic-statistical chart realisation algorithm combining (1) a theoretically grounded approach to syntax and semantic composition with (2) integrated language models for making choices among the options left open by the grammar. In so doing, it brings together the traditions of symbolic chart realisation (Kay, 1996; Carroll et al., 1999) and statistical realisation (Langkilde and Knight, 1998; Langkilde, 2000; Bangalore and Rambow, 2000; Langkilde-Geary, 2002). Another recent approach to combining these traditions appears in (Carroll and Oepen, 2005), where parse selection techniques are incorporated into an HPSG realiser.

In OpenCCG, the search for complete realisations makes use of n -gram language models and proceeds in one of two modes, *anytime* or *two-stage* (packing/unpacking). In the anytime mode, a best-first search is performed with a configurable time limit: the scores assigned by the n -gram model determine the order of the edges on the agenda, and thus have an impact on realisation speed. In the two-stage mode, a packed forest of all possible realisations is created in the first stage; in the second stage, the packed representation is unpacked in bottom-up fashion, with scores assigned to the edge for each sign as it is unpacked, much as in (Langkilde, 2000).

To realise a broad range of paraphrases, OpenCCG implements an algorithm for efficiently generating from disjunctive logical forms (LFs) (White, 2006a). A disjunctive LF represents the full set of possible syntactic paraphrases of a sentence: the differences may be subtle (e.g., choosing between *the design* or *it* as the subject), or may involve entirely different structures (e.g., *here we have a design in the classic style* vs. *this design is classic*). The algorithm uses packed representations similar to those initially proposed by Shemtov (1997), enabling it to run many times faster than sequential realisation of an equivalent set of non-disjunctive LFs.

The implementation described here makes use of the OpenCCG grammar developed as part of the COMIC multimodal dialogue system. This grammar was manually written with the aim of achieving very high quality. However, to streamline grammar development, the grammar was allowed to overgenerate in areas where rules are difficult to write and where n -gram models can be reliable; in particular, the grammar does not sufficiently constrain modifier order, which in the case of adverb placement especially can lead to a large number of possible orderings. To select preferred word orders among those allowed by the grammar for the input LF, we used a backoff 4-gram model trained on approximately 750 example target sentences, where certain words were replaced with their semantic classes (e.g. MANUFACTURER, COLOUR) for better generalisation, much as in (Oh and Rudnicky, 2002).

4 Anti-Repetition Methods

For both studies in this paper, we used OpenCCG to realise a range of texts describing and comparing bathroom-tile designs. The starting point for this implementation was the XSLT-based text planner from the COMIC system (Foster and White, 2004), which transforms sets of facts about tile designs into OpenCCG logical forms. We enhanced this text planner to produce disjunctive logical forms covering the full range of paraphrases permitted by the most recent version of the COMIC grammar, and then used OpenCCG realise those forms as text.

In the normal OpenCCG realisation process outlined above, corpus-based n -grams are used to select the single highest-scoring realisation for a given logical form. To allow the realiser to choose paraphrases other than the top-scoring one, we modified the realisation process in two ways: ϵ -best sampling and *anti-repetition scoring*.

ϵ -best sampling was implemented by creating the full set of possible surface realisations for a given disjunctive LF, and then randomly selecting one from the set whose n -gram score is within a given threshold of the top score. As the scores for sentences can vary by several orders of magnitude, the threshold was specified as a distance in log-10 space. Depending on the threshold value, this method can add more or less variation to the generated text. There is a danger that, if the threshold is too large and the grammar overgenerates, the output may include paraphrases that are dispreferred, or even ungrammatical.



Default (no anti-repetition methods)		With anti-repetition methods
This design is country. It is based on the Sandstein collection by Porcelaingres. The colours are brown, grey and black. There are geometric shapes on the decorative tiles.		Here is a design in the country style. It uses tiles from the Sandstein collection by Porcelaingres. It has brown, grey and black in the colour scheme. The decorative tiles have geometric shapes.
This design is also country. It is based on the Cardiff collection by Aparici. The colours are cream and dark red. It also has geometric shapes on the decorative tiles.		This one is also country. It draws from Cardiff, by Aparici. The colour scheme features cream and dark red. The decorative tiles also have geometric shapes.

Figure 1: Sample description sequence realised in both modes

Anti-repetition scoring was implemented as follows. First, for a proposed realisation, the number c of open-class words repeated from the preceding discourse was counted. This count was weighted: a word that appeared in the immediately preceding context received a full count of 1, one that appeared only in the context before that was weighted at 0.5, one from further back at 0.25, and so on. The repetition score r for a proposed realisation was then 10^{-p*c} , where p is the specific penalty value. This formula returns 1 if there are no repeated items, and returns a score that is linear in log space with the number of repeated items otherwise. The overall score for a proposed realisation was computed by multiplying r by the normal corpus-based n -gram score. In this way, preferences regarding word order and function words are still determined by the n -gram model, since the anti-repetition scorer only considers single open-class words.

5 Human Judgement Study

As a first test of the effectiveness of the two anti-repetition methods, we measured human judges' subjective responses to texts generated with and without these methods.

5.1 Materials and presentation

For this study, we used the text planner to create disjunctive logical forms for a set of eight description sequences. Each sequence consisted of four consecutive descriptions of tile patterns, using a fixed structure for all descriptions. We then used OpenCCG to generate text from these logical forms in two ways: in the default mode with no anti-repetition methods enabled, and with both ϵ -best sampling and anti-repetition scoring enabled, using a value of 20 for the parameter in each. For the anti-repetition scorer, each description as a whole provided the context for the sentences in the next: that

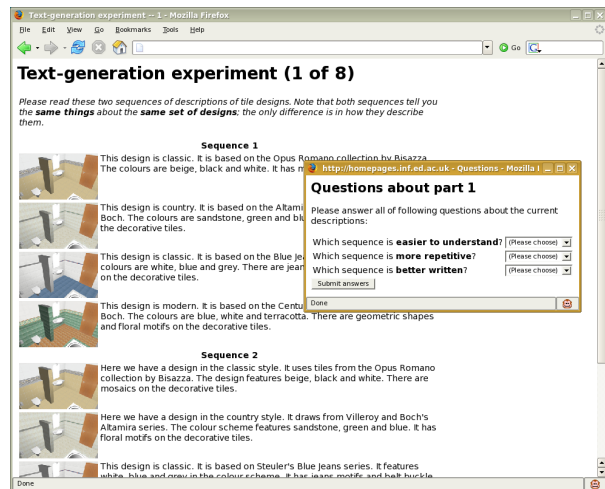


Figure 2: Evaluation interface

is, an entire set of sentences describing one design was realised, and then the words from all of those sentences were added to the context before the next description was processed. Figure 1 shows the first two descriptions of one of the generated sequences realised in both modes.

The experiment was run over the world-wide web and proceeded as follows. A participant was presented with both versions of each generated sequence in turn, in an individual randomly-chosen order. The order of presentation was counterbalanced so that each participant saw the default version first for four of the sequences, and the anti-repetition version first for the other four. A small thumbnail image of the tile design being described was shown beside each description. For each sequence, participants answered three forced-choice questions: which of the versions was (1) easier to understand, (2) more repetitive, and (3) better written. Figure 2 shows the user interface for this evaluation running in a web browser.

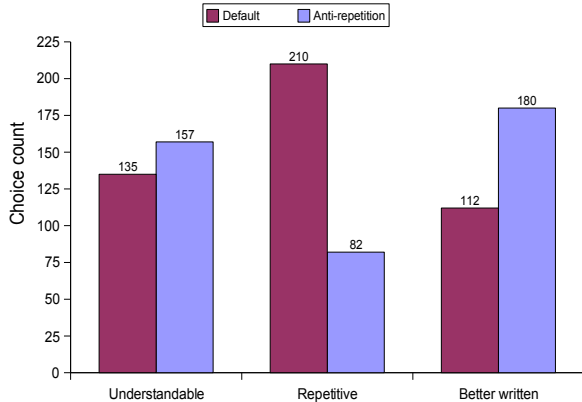


Figure 3: Results of the human evaluation

5.2 Participants and results

A total of 37 subjects took part in the evaluation study. All were native speakers of English; 20 were female, and 17 male. Since one subject answered half of the questions, this resulted in a total of 292 responses for each question.

The overall results are presented in Figure 3. For the understandability question, subjects chose the anti-repetition version in 157 cases (54%); this preference was not significant on a binomial test ($p \approx 0.2$). However, the responses to the other two questions did show significant preferences: subjects chose the default version as more repetitive 210 times (72%) and the anti-repetition version as better written 180 times (62%). Both of these preferences are significant at the $p < 0.0001$ level.

6 Exploring the Parameter Settings

The results of the user evaluation show that subjects found text generated with anti-repetition methods both less repetitive and better written. However, both methods depend on the value of a parameter: the threshold for ϵ -best sampling, and the repetition penalty for anti-repetition scoring. In the human evaluation, both parameters were set to the rather large value of 20. In this section, we explore the relative impact of the two methods by varying the configuration of the realiser and examining the generated texts for two factors: the variability across descriptions and the rate of dispreferred paraphrases.

For this experiment, we created the logical forms for a new set of 20 sequences of four descriptions, similar to those created for the human evaluation. For each sequence, we ran the realiser on all of the logical forms in turn, using all combinations of the

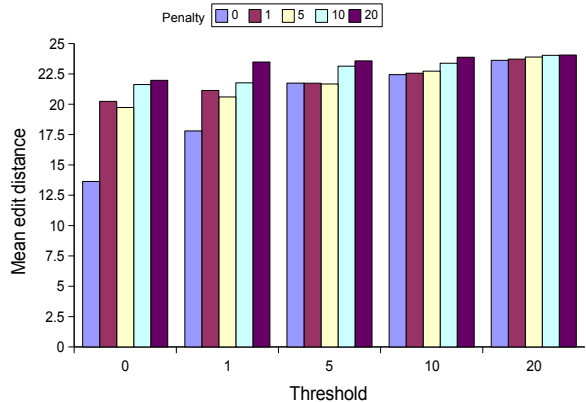


Figure 4: Mean edit distances

following values for each parameter: 0, 1, 5, 10, and 20. A threshold of 0 means that the realiser choose the highest-scoring result, while a repetition penalty of 0 amounts to no repetition penalty at all. When both parameters are set to 0, this corresponds to the default sequences from the human evaluation; when both parameters are 20, this corresponds to the anti-repetition sequences from that study.

As in the previous study, we realised all of the sentences for a given description and then added the results to the context for the anti-repetition scorer for the next descriptions. To compensate for any variability introduced into the process by the random choice in ϵ -best sampling, we realised the whole set of 20 sequences a total of six times.

6.1 Variability

To assess the variability in a generated description sequence, we computed the edit distance between all pairs of descriptions in the sequence; that is, the number of insertions, deletions, and replacements required to transform one description into another. Guégan and Hernandez (2006) used a similar edit-distance-based metric to detect parallelism in texts. The score for a sequence was the mean edit distance between all pairs of descriptions in the sequence, where a higher score indicates greater variability. As a concrete example, the edit distance between the two default descriptions in Figure 1 is 10, while the distance between the anti-repetition descriptions is 24; the mean edit distance for the sequences from the human evaluation was 13.4 for the default versions and 23.1 for the anti-repetition versions.

Figure 4 shows the mean edit distance for all settings of the parameters. Each set of five bars corresponds to a different setting of the threshold pa-

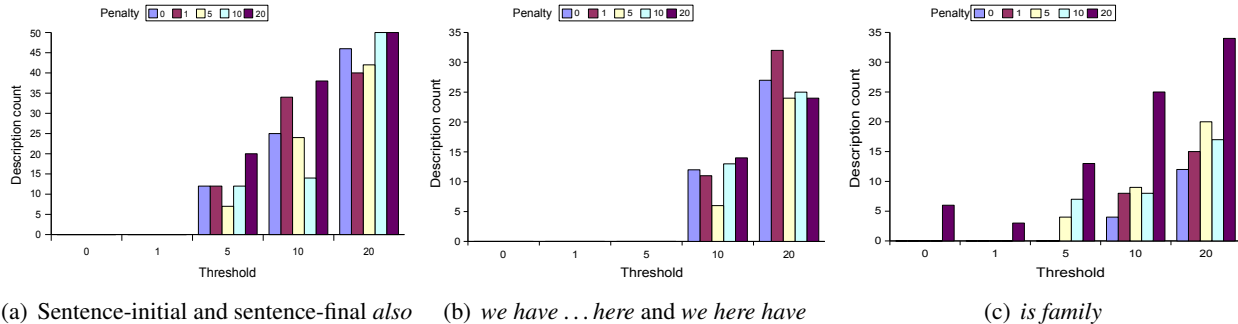


Figure 5: Counts for dispreferred paraphrases

parameter; within a set of bars, each shows the result with a different value for the penalty. To assess the significance of these results, we performed a linear regression, treating the values of each parameter as levels of an ordered factor. The resulting model explained approximately 70% of the total variance ($R^2 = 0.71$). The regression coefficients for each of the individual factors (threshold and penalty) were both significantly greater than 0 ($p < 0.0001$), showing that an increase in either tended to result in a corresponding increase in the edit distance. However, the regression coefficient for the interaction of the factors is negative (also $p < 0.0001$), indicating that the effect of the two methods is not simply additive.

6.2 Dispreferred paraphrases

To measure the rate of dispreferred paraphrases, we searched for specific word sequences that are permitted by the COMIC grammar, but that were deliberately not included in the OpenCCG n -gram models. Normally, the corpus-based n -grams ensure that such realisations are never included in the output; however, when the selection strategy is modified as described here, such word sequences can end up being selected. The occurrences of the following substrings were counted: sentence-initial and sentence-final *also*; *we here have ...* and *we have ... here* (instead of *here we have*); *is family* (instead of *is in the family style*).² In the anti-repetition descriptions used in the human evaluation, there was one instance each of *is family* and sentence-initial *also*.

Figure 5 shows the counts of dispreferred paraphrases under all of the parameter settings; again,

²Unlike *classic style*, *family style* is actually a noun-noun compound, but is not modelled that way in the grammar for uniformity. This means that the grammar also generates *is family*, which is odd, so n -grams were used to avoid this wording.

each group of bars corresponds to a different setting of the ϵ -best threshold, while each bar within the group represents a different value for the anti-repetition penalty. A total of 480 descriptions were generated under each combination of parameter settings: 20 sequences, each consisting of 4 descriptions, each generated 6 times. The count for each setting indicates the number of those descriptions that contained the specific substring. For example, 35 (7%) of the descriptions generated with both parameters set to 20 contained *is family* (Figure 5(c)). By inspection, it is clear that all dispreferred paraphrases tend to occur very infrequently at low parameter settings and to increase as the threshold increases; increasing the anti-repetition appears to have an effect only on *is family*.

To assess the significant factors for each of the dispreferred paraphrases, we analysed the influence of both parameters on the rate of that paraphrase by fitting a log-linear model to the contingency table of frequency counts for each of the paraphrase types; this type of model is suitable for use on count data and allows us to assess the influence of each of the factors on the counts in the table. The results confirm what is evident from the graph: increasing the threshold has a significant influence on the rate of all three of the paraphrases ($p < 0.0001$, ANOVA), while increasing the repetition penalty affects only the occurrence of *is family* (also $p < 0.0001$).

7 Discussion

The results of the user evaluation show that human judges strongly preferred the texts generated with the anti-repetition methods, even though the corpus-based n -gram score of such texts is lower than the score of the texts generated without such methods. This result agrees with the results of other recent studies that compared human preferences on gener-

ated output with the prediction of corpus-based similarity measures (e.g., Stent et al., 2005; Belz and Reiter, 2006; Foster and Oberlander, 2006). In all of these studies, human judges generally preferred outputs that incorporated more variation, even if the results were less similar to the corpus examples; on the other hand, corpus-based measures tended to favour output that did not diverge far, on average, from the corpus data.

By specifically concentrating on the effect of repetition in a discourse context, the results of the user study extend those of previous evaluations of the impact of variation in automatically-generated output, which generally presented the materials as a series of isolated examples. For example, Belz and Reiter (2006) evaluated a range of knowledge-based stochastic surface realisers. Their “greedy roulette” implementation, which selected generation rules based on corpus probabilities, had a similar effect on the generated texts as our variation methods: their implementation “will tend to use different words and phrases in different texts, whereas the other statistical generators will stick to those with the highest frequency.” This generator was penalised by automated evaluation measures because it tended to diverge from the corpus more than the others; however, the expert human judges ranked the output of this generator better than their bigram generator, though not as highly at their greedy one.

We considered two methods for avoiding repetition while generating text: ϵ -best sampling and anti-repetition scoring. These two methods were both straightforward to add to OpenCCG’s stochastic realisation process. Both had a significant effect on the variability across a sequence of descriptions, as measured by the mean edit distance between the elements of the sequence, although the effect of the two techniques was not additive. ϵ -best sampling also tended to increase the incidence of all dispreferred n -grams as the threshold value is increased, while anti-repetition scoring increased the rate only of the dispreferred n -grams that involved lexical choice.

If we compare the preferences in the user evaluation with the results of the automated studies, we see that the users tended to prefer the outputs that had higher variability. The materials generated for the user study happened to have very few dispreferred paraphrases—one instance each of *is family* and sentence-initial *also*—so it is difficult to draw definitive conclusions. The responses for the *also* description are similar to those on the entire set;

however, for the description with *is family*, the responses were significantly different. On this single item, 70% of the subjects chose the description generated without the anti-repetition methods (and therefore without the dispreferred paraphrase) as being better written; the responses on this question are significantly different than those on the rest of the items ($\chi^2 = 12.5$, $df = 1$, $p < 0.0001$). This suggests that, while the variability introduced by the anti-repetition methods is indeed appreciated by the human judges, there is nevertheless a real danger that departing too far from the corpus examples can lead to undesirable outputs.

8 Conclusions and future work

We have described two methods for enhancing the variation in the output of the OpenCCG surface realiser: ϵ -best sampling and anti-repetition scoring. In a human evaluation comparing text generated with and without these enhancements, subjects judged the versions generated with these methods to be better written and less repetitive significantly more often than the reverse, and did not report any difference in understandability between the versions. To our knowledge, this is the first study that specifically demonstrates the benefits of avoiding syntactic repetition as a discourse progresses.

When the impact of each of the two implemented methods on the generated text is examined, both have a similar effect on the variation across a sequence of descriptions. However the simpler ϵ -based technique is more prone to introducing dispreferred variants into the output, indicating that better results can be obtained using anti-repetition scoring with strict or no ϵ -based sampling. Using anti-repetition scoring also allows the anytime mode of the OpenCCG realiser to be employed.

In the human evaluation, participants were asked to give direct judgements on the quality of generated output presented as text: their responses indicated that they both were aware of and appreciated the variation in the output. In future evaluations, we would like to measure the impact of this type of variation on actual user interactions using tools such as subjective user satisfaction, objective dialog quality, and performance on a recall task.

An important consideration when automatically generating paraphrases is that any changes to the words or syntax should not alter the meaning; that is, in machine-translation terms, a paraphrase must be *adequate*. In our implementation, we ensured ad-

equacy through domain-specific text-planning rules that add options to a disjunctive logical form only if they are considered equivalent in the domain. It remains for future work to examine whether the engineering of such rules can be streamlined through automatic acquisition. Another question for future work is to investigate whether cases where repetition is useful can be identified, e.g. to achieve desired parallelism, and whether such insights can be incorporated into rules which restrict the paraphrase space accordingly.

Acknowledgements

This work was partly supported by the COMIC project (IST-2001-32311). Thanks to Jon Oberlander and the ENLG reviewers for useful comments.

References

- S. Bangalore and O. Rambow. 2000. Exploiting a probabilistic hierarchical model for generation. In *Proceedings, COLING-00*. ACL C00-1007.
- R. Barzilay and L. Lee. 2003. Learning to paraphrase: An unsupervised approach using multiple-sequence alignment. In *Proceedings, HLT-NAACL 2003*. ACL N03-1003.
- R. Barzilay and K. McKeown. 2001. Extracting paraphrases from a parallel corpus. In *Proceedings, ACL/EACL 2001*. ACL P01-1008.
- A. Belz and E. Reiter. 2006. Comparing automatic and human evaluation of NLG systems. In *Proceedings, EACL 2006*. ACL E06-1040.
- J. Carroll, A. Copestake, D. Flickinger, and V. Poznański. 1999. An efficient chart generator for (semi-) lexicalist grammars. In *Proceedings, EWNLG-99*.
- J. Carroll and S. Oepen. 2005. High efficiency realization for a wide-coverage unification grammar. In *Proceedings, IJCNLP-05*.
- K. van Deemter, E. Krahmer, and M. Theune. 2005. Real versus template-based natural language generation: A false opposition? *Computational Linguistics*, 31(1):15–24.
- M. E. Foster and J. Oberlander. 2006. Data-driven generation of emphatic facial displays. In *Proceedings, EACL 2006*. ACL E06-1045.
- M. E. Foster and M. White. 2004. Techniques for text planning with XSLT. In *Proceedings, NLPXML 2004*. ACL W04-0601.
- M. Guégan and N. Hernandez. 2006. Recognizing textual parallelisms with edit distance and similarity degree. In *Proceedings, EACL 2006*. ACL E06-1036.
- L. Iordanskaja, R. Kittredge, and A. Polguère. 1991. Lexical selection and paraphrase in a meaning-text generation model. In C. L. Paris, W. R. Swartout, and W. C. Mann, editors, *Natural Language Generation in Artificial Intelligence and Computational Linguistics*, pages 293–312. Kluwer.
- A. Isard, C. Brockmann, and J. Oberlander. 2006. Individuality and alignment in generated dialogues. In *Proceedings, INLG 2006*. ACL W06-1405.
- M. Kay. 1996. Chart generation. In *Proceedings, ACL-96*. ACL P96-1027.
- I. Langkilde. 2000. Forest-based statistical sentence generation. In *Proceedings, NAACL-00*. ACL A00-2023.
- I. Langkilde and K. Knight. 1998. Generation that exploits corpus-based statistical knowledge. In *Proceedings, COLING-ACL 1998*. ACL P98-1116.
- I. Langkilde-Geary. 2002. An empirical verification of coverage and correctness for a general-purpose sentence generator. In *Proceedings, INLG-02*.
- A. H. Oh and A. I. Rudnicky. 2002. Stochastic natural language generation for spoken dialog systems. *Computer, Speech & Language*, 16(3/4):387–407. doi:10.1016/S0885-2308(02)00012-8.
- B. Pang, K. Knight, and D. Marcu. 2003. Syntax-based alignment of multiple translations: Extracting paraphrases and generating new sentences. In *Proceedings, HLT-NAACL 2003*. ACL N03-1024.
- H. Shemtov. 1997. *Ambiguity Management in Natural Language Generation*. Ph.D. thesis, Stanford University.
- M. Steedman. 2000. *The Syntactic Process*. MIT Press.
- A. Stent, M. Marge, and M. Singhai. 2005. Evaluating evaluation methods for generation in the presence of variation. In *Computational Linguistics and Intelligent Text Processing*, volume 3406/2005 of *Lecture Notes in Computer Science*, pages 341–351. Springer. doi:10.1007/b105772.
- M. Stone, D. DeCarlo, I. Oh, C. Rodriguez, A. Stere, A. Lees, and C. Bregler. 2004. Speaking with hands: Creating animated conversational characters from recordings of human performance. *ACM Transactions on Graphics (SIGGRAPH)*, 23(3). doi:10.1145/1186562.1015753.
- M. White. 2005. Designing an extensible API for integrating language modeling and realization. In *Proceedings, ACL-05 Workshop on Software*.
- M. White. 2006a. CCG chart realization from disjunctive inputs. In *Proceedings, INLG 2006*. ACL W06-1403.
- M. White. 2006b. Efficient realization of coordinate structures in Combinatory Categorical Grammar. *Research on Language and Computation*, 4(1):39–75. doi:10.1007/s11168-006-9010-2.