Learning to Say It Well:
Reranking Realizations by Predicted Synthesis Quality

Crystal Nakatsu and **Michael White**
Department of Linguistics
The Ohio State University
Columbus, OH 43210 USA
{cnakatsu,mwhite}@ling.ohio-state.edu

Abstract

This paper presents a method for adapting a language generator to the strengths and weaknesses of a synthetic voice, thereby improving the naturalness of synthetic speech in a spoken language dialogue system. The method trains a discriminative reranker to select paraphrases that are predicted to sound natural when synthesized. The ranker is trained on realizer and synthesizer features in supervised fashion, using human judgements of synthetic voice quality on a sample of the paraphrases representative of the generator's capability. Results from a cross-validation study indicate that discriminative paraphrase reranking can achieve substantial improvements in naturalness on average, ameliorating the problem of highly variable synthesis quality typically encountered with today's unit selection synthesizers.

1 Introduction

Unit selection synthesis[1]—a technique which concatenates segments of natural speech selected from a database—has been found to be capable of producing high quality synthetic speech, especially for utterances that are similar to the speech in the database in terms of style, delivery, and coverage (Black and Lenzo, 2001). In particular, in the limited domain of a spoken language dialogue system, it is possible to achieve highly natural synthesis with a purpose-built voice (Black and Lenzo, 2000). However, it can be difficult to develop a synthetic voice for a dialogue system that produces natural speech completely reliably, and thus in practice output quality can be quite variable. Two important factors in this regard are the labeling process for the speech database and the direction of the dialogue system's further development, after the voice has been built: when labels are assigned fully automatically to the recorded speech, label boundaries may be inaccurate, leading to unnatural sounding joins in speech output; and when further system development leads to the generation of utterances that are less like those in the recording script, such utterances must be synthesized using smaller units with more joins between them, which can lead to a considerable dropoff in quality.

As suggested by Bulyko and Ostendorf (2002), one avenue for improving synthesis quality in a dialogue system is to have the system choose what to say in part by taking into account what is likely to sound natural when synthesized. The idea is to take advantage of the generator's periphrastic ability:[2] given a set of generated paraphrases that suitably express the desired content in the dialogue context, the system can select the specific paraphrase to use as its response according to the predicted quality of the speech synthesized for that paraphrase. In this way, if there are significant differences in the predicted synthesis quality for the various paraphrases—and if these predictions are generally borne out—then, by selecting paraphrases with high predicted synthesis quality, the dialogue system (as a whole) can more reliably produce natural sounding speech.

In this paper, we present an application of dis-

[1] See e.g. (Hunt and Black, 1996; Black and Taylor, 1997; Beutnagel et al., 1999).

[2] See e.g. (Iordanskaja et al., 1991; Langkilde and Knight, 1998; Barzilay and McKeown, 2001; Pang et al., 2003) for discussion of paraphrase in generation.

Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL, pages 1113–1120,
Sydney, July 2006.

criminative reranking to the task of adapting a language generator to the strengths and weaknesses of a particular synthetic voice. Our method involves training a reranker to select paraphrases that are predicted to sound natural when synthesized, from the N-best realizations produced by the generator. The ranker is trained in supervised fashion, using human judgements of synthetic voice quality on a representative sample of the paraphrases. In principle, the method can be employed with any speech synthesizer. Additionally, when features derived from the synthesizer's unit selection search can be made available, further quality improvements become possible.

The paper is organized as follows. In Section 2, we review previous work on integrating choice in language generation and speech synthesis, and on learning discriminative rerankers for generation. In Section 3, we present our method. In Section 4, we describe a cross-validation study whose results indicate that discriminative paraphrase reranking can achieve substantial improvements in naturalness on average. Finally, in Section 5, we conclude with a summary and a discussion of future work.

2 Previous Work

Most previous work on integrating language generation and synthesis, e.g. (Davis and Hirschberg, 1988; Prevost and Steedman, 1994; Hitzeman et al., 1998; Pan et al., 2002), has focused on how to use the information present in the language generation component in order to specify contextually appropriate intonation for the speech synthesizer to target. For example, syntactic structure, information structure and dialogue context have all been argued to play a role in improving prosody prediction, compared to unrestricted text-to-speech synthesis. While this topic remains an important area of research, our focus is instead on a different opportunity that arises in a dialogue system, namely, the possibility of choosing the exact wording and prosody of a response according to how natural it is likely to sound when synthesized.

To our knowledge, Bulyko and Ostendorf (2002) were the first to propose allowing the choice of wording and prosody to be jointly determined by the language generator and speech synthesizer. In their approach, a template-based generator passes a prosodically annotated word network to the speech synthesizer, rather than a single text string (or prosodically annotated text string). To perform the unit selection search on this expanded input efficiently, they employ weighted finite-state transducers, where each step of network expansion is then followed by minimization. The weights are determined by concatenation (join) costs, relative frequencies (negative log probabilities) of the word sequences, and prosodic prediction costs, for cases where the prosody is not determined by the templates. In a perception experiment, they demonstrated that by expanding the space of candidate responses, their system achieved higher quality speech output.

Following (Bulyko and Ostendorf, 2002), Stone et al. (2004) developed a method for jointly determining wording, speech and gesture. In their approach, a template-based generator produces a word lattice with intonational phrase breaks. A unit selection algorithm then searches for a low-cost way of realizing a path through this lattice that combines captured motion samples with recorded speech samples to create coherent phrases, blending segments of speech and motion together phrase-by-phrase into extended utterances. Video demonstrations indicate that natural and highly expressive results can be achieved, though no human evaluations are reported.

In an alternative approach, Pan and Weng (2002) proposed integrating instance-based realization and synthesis. In their framework, sentence structure, wording, prosody and speech waveforms from a domain-specific corpus are simultaneously reused. To do so, they add prosodic and acoustic costs to the insertion, deletion and replacement costs used for instance-based surface realization. Their contribution focuses on how to design an appropriate speech corpus to facilitate an integrated approach to instance-based realization and synthesis, and does not report evaluation results.

A drawback of these approaches to integrating choice in language generation and synthesis is that they cannot be used with most existing speech synthesizers, which do not accept (annotated) word lattices as input. In contrast, the approach we introduce here can be employed with any speech synthesizer in principle. All that is required is that the language generator be capable of producing N-best outputs; that is, the generator must be able to construct a set of suitable paraphrases expressing the desired content, from which the top N realizations can be selected for reranking according to their predicted synthesis quality. Once the realizations have been reranked, the top scoring realization can be sent to the synthesizer as usual. Alternatively, when features derived from the synthesizer's unit selection search can be made available—and if the time demands of the dialogue system permit—several of the top scoring reranked realizations can be sent to the synthesizer, and the resulting utterances can be rescored with the extended feature set.

Our reranking approach has been inspired by previous work on reranking in parsing and generation, especially (Collins, 2000) and (Walker et al., 2002). As in Walker et al.'s (2002) method for training a sentence plan ranker, we use our generator to produce a representative sample of paraphrases and then solicit human judgements of their naturalness to use as data for training the ranker. This method is attractive when there is no suitable corpus of naturally occurring dialogues available for training purposes, as is often the case for systems that engage in human-computer dialogues that differ substantially from human-human ones. The primary difference between Walker et al.'s work and ours is that theirs examines the impact on text quality of sentence planning decisions such as aggregation, whereas ours focuses on the impact of the lexical and syntactic choice at the surface realization level on speech synthesis quality, according to the strengths and weaknesses of a particular synthetic voice.

3 Reranking Realizations by Predicted Synthesis Quality

3.1 Generating Alternatives

Our experiments with integrating language generation and synthesis have been carried out in the context of the COMIC[3] multimodal dialogue system (den Os and Boves, 2003). The COMIC system adds a dialogue interface to a CAD-like application used in sales situations to help clients redesign their bathrooms. The input to the system includes speech, handwriting, and pen gestures; the output combines synthesized speech, an animated talking head, deictic gestures at on-screen objects, and direct control of the underlying application.

Drawing on the materials used in (Foster and White, 2005) to evaluate adaptive generation in COMIC, we selected a sample of 104 sentences from 38 different output turns across three dialogues. For each sentence in the set, a variant was included that expressed the same content adapted to a different user model or adapted to a different dialogue history. For example, a description of a certain design's colour scheme for one user might be phrased as *As you can see, the tiles have a blue and green colour scheme*, whereas a variant expression of the same content for a different user could be *Although the tiles have a blue colour scheme, the design does also feature green*, if the user disprefers blue.

In COMIC, the sentence planner uses XSLT to generate disjunctive logical forms (LFs), which specify a range of possible paraphrases in a nested free-choice form (Foster and White, 2004). Such disjunctive LFs can be efficiently realized using the OpenCCG realizer (White, 2004; White, 2006b; White, 2006a). Note that for the experiments reported here, we manually augmented the disjunctive LFs for the 104 sentences in our sample to make greater use of the periphrastic capabilities of the COMIC grammar; it remains for future work to augment the COMIC sentence planner produce these more richly disjunctive LFs automatically.

OpenCCG includes an extensible API for integrating language modeling and realization. To select preferred word orders, from among all those allowed by the grammar for the input LF, we used a backoff trigram model trained on approximately 750 example target sentences, where certain words were replaced with their semantic classes (e.g. MANUFACTURER, COLOUR) for better generalization. For each of the 104 sentences in our sample, we performed 25-best realization from the disjunctive LF, and then randomly selected up to 12 different realizations to include in our experiments based on a simulated coin flip for each realization, starting with the top-scoring one. We used this procedure to sample from a larger portion of the N-best realizations, while keeping the sample size manageable.

Figure 1 shows an example of 12 paraphrases for a sentence chosen for inclusion in our sample. Note that the realizations include words with pitch accent annotations as well as boundary tones as separate, punctuation-like words. Generally the

[3] COnversational Multimodal Interaction with Computers, `http://www.hcrc.ed.ac.uk/comic/`.

- this$_{H*}$ design$_{H*}$ uses tiles from Villeroy_and_Boch$_{H*}$'s Funny_Day$_{H*}$ collection LL% .
- this$_{H*}$ design$_{H*}$ is based_on the Funny_Day$_{H*}$ collection by Villeroy_and_Boch$_{H*}$ LL% .
- this$_{H*}$ design$_{H*}$ is based_on Funny_Day$_{H*}$ LL% , by Villeroy_and_Boch$_{H*}$ LL% .
- this$_{H*}$ design$_{H*}$ draws from the Funny_Day$_{H*}$ collection by Villeroy_and_Boch$_{H*}$ LL% .
- this$_{H*}$ one draws from Funny_Day$_{H*}$ LL% , by Villeroy_and_Boch$_{H*}$ LL% .
- here$_{L+H*}$ LH% we have a design that is based_on the Funny_Day$_{H*}$ collection by Villeroy_and_Boch$_{H*}$ LL% .
- this$_{H*}$ design$_{H*}$ draws from Villeroy_and_Boch$_{H*}$'s Funny_Day$_{H*}$ series LL% .
- here is a design that draws from Funny_Day$_{H*}$ LL% , by Villeroy_and_Boch$_{H*}$ LL% .
- this$_{H*}$ one draws from Villeroy_and_Boch$_{H*}$'s Funny_Day$_{H*}$ collection LL% .
- this$_{H*}$ draws from the Funny_Day$_{H*}$ collection by Villeroy_and_Boch$_{H*}$ LL% .
- this$_{H*}$ one draws from the Funny_Day$_{H*}$ collection by Villeroy_and_Boch$_{H*}$ LL% .
- here is a design that draws from Villeroy_and_Boch$_{H*}$'s Funny_Day$_{H*}$ collection LL% .

Figure 1: Example of sampled periphrastic alternatives for a sentence.

quality of the sampled paraphrases is very high, only occasionally including dispreferred word orders such as *We here have a design in the family style*, where *here* is in medial position rather than fronted.[4]

3.2 Synthesizing Utterances

For synthesis, OpenCCG's output realizations are converted to APML,[5] a markup language which allows pitch accents and boundary tones to be specified, and then passed to the Festival speech synthesis system (Taylor et al., 1998; Clark et al., 2004). Festival uses the prosodic markup in the text analysis phase of synthesis in place of the structures that it would otherwise have to predict from the text. The synthesiser then uses the context provided by the markup to enforce the selection of suitable units from the database.

A custom synthetic voice for the COMIC system was developed, as follows. First, a domain-specific recording script was prepared by selecting about 150 sentences from the larger set of target sentences used to train the system's n-gram model. The sentences were greedily selected with the goals of ensuring that (i) all words (including proper names) in the target sentences appeared at least once in the record script, and (ii) all bigrams at the level of semantic classes (e.g. MANUFACTURER, COLOUR) were covered as well. For the cross-validation study reported in the next section, we also built a trigram model on the words in the domain-specific recording script, *without* replacing any words with semantic classes, so that we could examine whether the more frequent occurrence of the specific words and phrases in this part of the script is predictive of synthesis quality.

The domain-specific script was augmented with a set of 600 newspaper sentences selected for diphone coverage. The newspaper sentences make it possible for the voice to synthesize words outside of the domain-specific script, though not necessarily with the same quality. Once these scripts were in place, an amateur voice talent was recorded reading the sentences in the scripts during two recording sessions. Finally, after the speech files were semi-automatically segmented into individual sentences, the speech database was constructed, using fully automatic labeling.

We have found that the utterances synthesized with the COMIC voice vary considerably in their naturalness, due to two main factors. First, the system underwent further development after the voice was built, leading to the addition of a variety of new phrases to the system's repertoire, as well as many extra proper names (and their pronunciations); since these names and phrases usually require going outside of the domain-specific part of the speech database, they often (though not always) exhibit a considerable dropoff in synthesis quality.[6] And second, the boundaries of the automatically assigned unit labels were not always accurate, leading to problems with unnatural joins and reduced intelligibility. To improve the reliability of the COMIC voice, we could have recorded more speech, or manually corrected label bound-

[4] In other examples medial position is preferred, e.g. *This design here is in the family style*.

[5] Affective Presentation Markup Language; see `http://www.cstr.ed.ac.uk/projects/festival/apml.html`.

[6] Note that in the current version of the system, proper names are always required parts of the output, and thus the discriminative reranker cannot learn to simply choose paraphrases that leave out problematic names.

aries; the goal of this paper is to examine whether the naturalness of a dialogue system's output can be improved in a less labor-intensive way.

3.3 Rating Synthesis Quality

To obtain data for training our realization reranker, we solicited judgements of the naturalness of the synthesized speech produced by Festival for the utterances in our sample COMIC corpus. Two judges (the first two authors) provided judgements on a 1–7 point scale, with higher scores representing more natural synthesis. Ratings were gathered using WebExp2,[7] with the periphrastic alternatives for each sentence presented as a group in a randomized order. Note that for practical reasons, the utterances were presented out of the dialogue context, though both judges were familiar with the kinds of dialogues that the COMIC system is capable of.

Though the numbers on the seven point scale were not assigned labels, they were roughly taken to be "horrible," "poor," "fair," "ok," "good," "very good" and "perfect." The average assigned rating across all utterances was 4.05 ("ok"), with a standard deviation of 1.56. The correlation between the two judges' ratings was 0.45, with one judge's ratings consistently higher than the other's.

Some common problems noted by the judges included slurred words, especially *the* sometimes sounding like *ther* or even *their*; clipped words, such as *has* shortened at times to the point of sounding like *is*, or *though* clipped to unintelligibility; unnatural phrasing or emphasis, e.g. occasional pauses before a possessive *'s*, or words such as *style* sounding emphasized when they should be deaccented; unnatural rate changes; "choppy" speech from poor joins; and some unintelligible proper names.

3.4 Ranking

While Collins (2000) and Walker et al. (2002) develop their rankers using the RankBoost algorithm (Freund et al., 1998), we have instead chosen to use Joachims' (2002) method of formulating ranking tasks as Support Vector Machine (SVM) constraint optimization problems.[8] This choice has been motivated primarily by convenience, as Joachims' SVM^{light} package is easy to use; we leave it for future work to compare the performance of RankBoost and SVM^{light} on our ranking task.

The ranker takes as input a set of paraphrases that express the desired content of each sentence, optionally together with synthesized utterances for each paraphrase. The output is a ranking of the paraphrases according to the predicted naturalness of their corresponding synthesized utterances. Ranking is more appropriate than classification for our purposes, as naturalnesss is a graded assessment rather than a categorical one.

To encode the ranking task as an SVM constraint optimization problem, each paraphrase j of a sentence i is represented by a feature vector $\Phi(s_{ij}) = \langle f_1(s_{ij}), \ldots, f_m(s_{ij}) \rangle$, where m is the number of features. In the training data, the feature vectors are paired with the average value of their corresponding human judgements of naturalness. From this data, ordered pairs of paraphrases (s_{ij}, s_{ik}) are derived, where s_{ij} has a higher naturalness rating than s_{ik}. The constraint optimization problem is then to derive a parameter vector $\vec{w}$ that yields a ranking score function $\vec{w} \cdot \Phi(s_{ij})$ which minimizes the number of pairwise ranking violations. Ideally, for every ordered pair (s_{ij}, s_{ik}), we would have $\vec{w} \cdot \Phi(s_{ij}) > \vec{w} \cdot \Phi(s_{ik})$; in practice, it is often impossible or intractable to find such a parameter vector, and thus slack variables are introduced that allow for training errors. A parameter to the algorithm controls the trade-off between ranking margin and training error.

In testing, the ranker's accuracy can be determined by comparing the ranking scores for every ordered pair (s_{ij}, s_{ik}) in the test data, and determining whether the actual preferences are borne out by the predicted preference, i.e. whether $\vec{w} \cdot \Phi(s_{ij}) > \vec{w} \cdot \Phi(s_{ik})$ as desired. Note that the ranking scores, unlike the original ratings, do not have any meaning in the absolute sense; their import is only to order alternative paraphrases by their predicted naturalness.

In our ranking experiments, we have used SVM^{light} with all parameters set to their default values.

3.5 Features

Table 1 shows the feature sets we have investigated for reranking, distinguished by the availability of the features and the need for discriminative training. The first row shows the feature sets that are

[7] `http://www.hcrc.ed.ac.uk/web_exp/`

[8] See (Barzilay and Lapata, 2005) for another application of SVM ranking in generation, namely to the task of ranking alternative text orderings for local coherence.

Table 1: Feature sets for reranking.

	Discriminative	
Availability	*no*	*yes*
Realizer	NGRAMS	WORDS
Synthesizer	COSTS	ALL

available to the realizer. There are two n-gram models that can be used to directly rank alternative realizations: NGRAM-1, the language model used in COMIC, and NGRAM-2, the language model derived from the domain-specific recording script; for feature values, the negative logarithms are used. There are also two WORDS feature sets (shown in the second column): WORDS-BI, which includes NGRAMS plus a feature for every possible unigram and bigram, where the value of the feature is the count of the unigram or bigram in a given realization; and WORDS-TRI, which includes all the features in WORDS-BI, plus a feature for every possible trigram. The second row shows the feature sets that require information from the synthesizer. The COSTS feature set includes NGRAMS plus the total join and target costs from the unit selection search. Note that a weighted sum of these costs could be used to directly rerank realizations, in much the same way as relative frequencies and concatenation costs are used in (Bulyko and Ostendorf, 2002); in our experiments, we let SVM^{light} determine how to weight these costs. Finally, there are two ALL feature sets: ALL-BI includes NGRAMS, WORDS-BI and COSTS, plus features for every possible phone and diphone, and features for every specific unit in the database; ALL-TRI includes NGRAMS, WORDS-TRI, COSTS, and a feature for every phone, diphone and triphone, as well as specific units in the database. As with WORDS, the value of a feature is the count of that feature in a given synthesized utterance.

4 Cross-Validation Study

To train and test our ranker on our feature sets, we partitioned the corpus into 10 folds and performed 10-fold cross-validation. For each fold, 90% of the examples were used for training the ranker and the remaining unseen 10% were used for testing. The folds were created by randomly choosing from among the sentence groups, resulting in all of the paraphrases for a given sentence occurring in the same fold, and each occurring exactly once in the testing set as a whole.

Table 2: Comparison of results for differing feature sets, topline and baseline.

Features	Mean Score	SD	Accuracy (%)
BEST	5.38	1.11	100.0
WORDS-TRI	4.95	1.24	77.3
ALL-BI	4.95	1.24	77.9
ALL-TRI	4.90	1.25	78.0
WORDS-BI	4.86	1.28	76.8
COSTS	4.69	1.27	68.2
NGRAM-2	4.34	1.38	56.2
NGRAM-1	4.30	1.29	53.3
RANDOM	4.11	1.22	50.0

We evaluated the performance of our ranker by determining the average score of the best ranked paraphrase for each sentence, under each of the following feature combinations: NGRAM-1, NGRAM-2, COSTS, WORDS-BI, WORDS-TRI, ALL-BI, and ALL-TRI. Note that since we used the human ratings to calculate the score of the highest ranked utterance, the score of the highest ranked utterance cannot be higher than that of the highest human-rated utterance. Therefore, we effectively set the human ratings as the topline (BEST). For the baseline, we randomly chose an utterance from among the alternatives, and used its associated score. In 15 tests generating the random scores, our average scores ranged from 3.88–4.18. We report the median score of 4.11 as the average for the baseline, along with the mean of the topline and each of the feature subsets, in Table 2.

We also report the ordering accuracy of each feature set used by the ranker in Table 2. As mentioned in Section 3.4, the ordering accuracy of the ranker using a given feature set is determined by c/N, where c is the number of correctly ordered pairs (of each paraphrase, not just the top ranked one) produced by the ranker, and N is the total number of human-ranked ordered pairs.

As Table 2 indicates, the mean of BEST is 5.38, whereas our ranker using WORDS-TRI features achieves a mean score of 4.95. This is a difference of 0.42 on a seven point scale, or only a 6% difference. The ordering accuracy of WORDS-TRI is 77.3%.

We also measured the improvement of our ranker with each feature set over the random baseline as a percentage of the maximum possible gain (which would be to reproduce the human topline). The results appear in Figure 2. As the

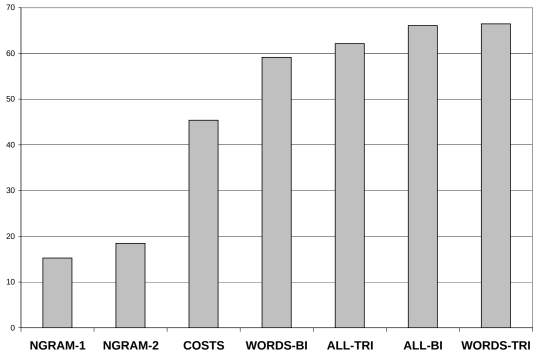

Figure 2: Improvement as a percentage of the maximum possible gain over the random baseline.

figure indicates, the maximum possible gain our ranker achieves over the baseline is 66% (using the WORDS-TRI or ALL-BI feature set) . By comparison, NGRAM-1 and NGRAM-2 achieve less than 20% of the possible gain.

To verify our main hypothesis that our ranker would significantly outperform the baselines, we computed paired one-tailed t-tests between WORDS-TRI and RANDOM ($t = 2.4$, $p < 8.9x10^{-13}$), and WORDS-TRI and NGRAM-1 ($t = 1.4$, $p < 4.5x10^{-8}$). Both differences were highly significant. We also performed seven post-hoc comparisons using two-tailed t-tests, as we did not have an *a priori* expectation as to which feature set would work better. Using the Bonferroni adjustment for multiple comparisons, the p-value required to achieve an overall level of significance of 0.05 is 0.007. In the first post-hoc test, we found a significant difference between BEST and WORDS-TRI ($t = 8.0$,$p < 1.86x10^{-12}$), indicating that there is room for improvement of our ranker. However, in considering the top scoring feature sets, we did not find a significant difference between WORDS-TRI and WORDS-BI ($t = 2.3, p < 0.022$), from which we infer that the difference among all of WORDS-TRI, ALL-BI, ALL-TRI and WORDS-BI is not significant also. This suggests that the synthesizer features have no substantial impact on our ranker, as we would expect ALL-TRI to be significantly higher than WORDS-TRI if so. However, since COSTS does significantly improve upon NGRAM2 ($t = 3.5$, $p < 0.001$), there is some value to the use of synthesizer features in the absence of WORDS. We also looked at the comparison for the WORDS models and COSTS. While WORDS-BI did not perform significantly better than COSTS ($t = 2.3$, $p < 0.025$), the added trigrams in WORDS-TRI did improve ranker performance significantly over COSTS ($t = 3.7$, $p < 3.29x10^{-4}$). Since COSTS ranks realizations in the much the same way as (Bulyko and Ostendorf, 2002), the fact that WORDS-TRI outperforms COSTS indicates that our discriminative reranking method can significantly improve upon their non-discriminative approach.

5 Conclusions

In this paper, we have presented a method for adapting a language generator to the strengths and weaknesses of a particular synthetic voice by training a discriminative reranker to select paraphrases that are predicted to sound natural when synthesized. In contrast to previous work on this topic, our method can be employed with any speech synthesizer in principle, so long as features derived from the synthesizer's unit selection search can be made available. In a case study with the COMIC dialogue system, we have demonstrated substantial improvements in the naturalness of the resulting synthetic speech, achieving two-thirds of the maximum possible gain, and raising the average rating from "ok" to "good." We have also shown that in this study, our discriminative method significantly outperforms an approach that performs selection based solely on corpus frequencies together with target and join costs.

In future work, we intend to verify the results of our cross-validation study in a perception experiment with naïve subjects. We also plan to investigate whether additional features derived from the synthesizer can better detect unnatural pauses or changes in speech rate, as well as F0 contours that fail to exhibit the targeting accenting pattern. Finally, we plan to examine whether gains in quality can be achieved with an off-the-shelf, general purpose voice that are similar to those we have observed using COMIC's limited domain voice.

Acknowledgements

We thank Mary Ellen Foster, Eric Fosler-Lussier and the anonymous reviewers for helpful comments and discussion.

References

Regina Barzilay and Mirella Lapata. 2005. Modeling local coherence: An entity-based approach. In *Pro-*

ceedings of the 43rd Annual Meeting of the Association for Computational Linguistics, Ann Arbor.

Regina Barzilay and Kathleen McKeown. 2001. Extracting paraphrases from a parallel corpus. In *Proc. ACL/EACL*.

M. Beutnagel, A. Conkie, J. Schroeter, Y. Stylianou, and A. Syrdal. 1999. The AT&T Next-Gen TTS system. In *Joint Meeting of ASA, EAA, and DAGA*.

Alan Black and Kevin Lenzo. 2000. Limited domain synthesis. In *Proceedings of ICSLP2000*, Beijing, China.

Alan Black and Kevin Lenzo. 2001. Optimal data selection for unit selection synthesis. In *4th ISCA Speech Synthesis Workshop*, Pitlochry, Scotland.

Alan Black and Paul Taylor. 1997. Automatically clustering similar units for unit selection in speech synthesis. In *Eurospeech '97*.

Ivan Bulyko and Mari Ostendorf. 2002. Efficient integrated response generation from multiple targets using weighted finite state transducers. *Computer Speech and Language*, 16:533–550.

Robert A.J. Clark, Korin Richmond, and Simon King. 2004. Festival 2 – build your own general purpose unit selection speech synthesiser. In *5th ISCA Speech Synthesis Workshop*, pages 173–178, Pittsburgh, PA.

Michael Collins. 2000. Discriminative reranking for natural language parsing. In *Proc. ICML*.

James Raymond Davis and Julia Hirschberg. 1988. Assigning intonational features in synthesized spoken directions. In *Proc. ACL*.

Els den Os and Lou Boves. 2003. Towards ambient intelligence: Multimodal computers that understand our intentions. In *Proc. eChallenges-03*.

Mary Ellen Foster and Michael White. 2004. Techniques for Text Planning with XSLT. In *Proc. 4th NLPXML Workshop*.

Mary Ellen Foster and Michael White. 2005. Assessing the impact of adaptive generation in the COMIC multimodal dialogue system. In *Proc. IJCAI-05 Workshop on Knowledge and Representation in Practical Dialogue Systems*.

Y. Freund, R. Iyer, R. E. Schapire, and Y. Singer. 1998. An efficient boosting algorithm for combining preferences. In *Machine Learning: Proc. of the Fifteenth International Conference*.

Janet Hitzeman, Alan W. Black, Chris Mellish, Jon Oberlander, and Paul Taylor. 1998. On the use of automatically generated discourse-level information in a concept-to-speech synthesis system. In *Proc. ICSLP-98*.

A. Hunt and A. Black. 1996. Unit selection in a concatenative speech synthesis system using a large speech database. In *Proc. ICASSP-96*, Atlanta, Georgia.

Lidija Iordanskaja, Richard Kittredge, and Alain Polgúere. 1991. Lexical selection and paraphrase in a meaning-text generation model. In Cécile L. Paris, William R. Swartout, and William C. Mann, editors, *Natural Language Generation in Artificial Intelligence and Computational Linguistics*, pages 293–312. Kluwer.

Thorsten Joachims. 2002. Optimizing search engines using clickthrough data. In *Proc. KDD*.

Irene Langkilde and Kevin Knight. 1998. Generation that exploits corpus-based statistical knowledge. In *Proc. COLING-ACL*.

Shimei Pan and Wubin Weng. 2002. Designing a speech corpus for instance-based spoken language generation. In *Proc. of the International Natural Language Generation Conference (INLG-02)*.

Shimei Pan, Kathleen McKeown, and Julia Hirschberg. 2002. Exploring features from natural language generation for prosody modeling. *Computer Speech and Language*, 16:457–490.

Bo Pang, Kevin Knight, and Daniel Marcu. 2003. Syntax-based alignment of multiple translations: Extracting paraphrases and generating new sentences. In *Proc. HLT/NAACL*.

Scott Prevost and Mark Steedman. 1994. Specifying intonation from context for speech synthesis. *Speech Communication*, 15:139–153.

Matthew Stone, Doug DeCarlo, Insuk Oh, Christian Rodriguez, Adrian Stere, Alyssa Lees, and Chris Bregler. 2004. Speaking with hands: Creating animated conversational characters from recordings of human performance. *ACM Transactions on Graphics (SIGGRAPH)*, 23(3).

P. Taylor, A. Black, and R. Caley. 1998. The architecture of the the Festival speech synthesis system. In *Third International Workshop on Speech Synthesis, Sydney, Australia*.

Marilyn A. Walker, Owen C. Rambow, and Monica Rogati. 2002. Training a sentence planner for spoken dialogue using boosting. *Computer Speech and Language*, 16:409–433.

Michael White. 2004. Reining in CCG Chart Realization. In *Proc. INLG-04*.

Michael White. 2006a. CCG chart realization from disjunctive logical forms. In *Proc. INLG-06*. To appear.

Michael White. 2006b. Efficient Realization of Coordinate Structures in Combinatory Categorial Grammar. *Research on Language & Computation*, online first, March.