Unsupervised features from text for speech synthesis in a speech-to-speech translation system

Oliver Watts[1], *Bowen Zhou*[2]

[1]Centre for Speech Technology Research, University of Edinburgh, UK
[2]IBM T. J. Watson Research Center, Yorktown Heights, NY 10598, USA

O.S.Watts@sms.ed.ac.uk, zhou@us.ibm.com

Abstract

We explore the use of linguistic features for text to speech (TTS) conversion in the context of a speech-to-speech translation system that can be extracted from unannotated text in an unsupervised, language-independent fashion. The features are intended to act as surrogates for conventional part of speech (POS) features. Unlike POS features, the experimental features assume only the availability of tools and data that must already be in place for the construction of other components of the translation system, and can therefore be used for the TTS module without incurring additional TTS-specific costs. We here describe the use of the experimental features in a speech synthesiser, using six different configurations of the system to allow the comparison of the proposed features with conventional, knowledge-based POS features. We present results of objective and subjective evaluations of the usefulness of the new features.
Index Terms: speech synthesis

1. Background

The porting of a speech-to-speech translation system to a new language pair requires the collection of data in the relevant languages. A conventional text-to-speech (TTS) component for such a system will ideally be able to predict linguistic features from text. The part of speech (POS) of a word and neighbouring words, for example, can be useful in predicting the acoustic realization of that word. However, manually annotated data are needed for training the necessary classifiers (e.g. the POS tagger), and the collection of such specialized resources where none are readily available is expensive and time-consuming, and will thus slow the porting of the system to the target language pair.

The work presented here explores the use of features for TTS that can be extracted from unannotated text in an unsupervised fashion, and which assume only the availability of tools and data that must already be in place for the construction of the other components of a speech-to-speech translation system (i.e. speech recognition and machine translation modules). Such features can therefore be used for the TTS module without incurring additional TTS-specific costs.

2. Data, Training Procedure, and Features Used in the Systems

All systems were built with approximately 2 hours of English read news data from a purpose-built commercial TTS database; audio and annotation for 150 sentences were set aside for use in objective and subjective evaluation. We chose to work with English for these experiments because POS annotation is readily available for the data-set, and thus allows controlled comparison of conventional POS features with automatically obtained ones. The proposed features, however, can be extracted in a language-independent way without requiring specially annotated data.

Six configurations of a Hidden Markov Model (HMM) based synthesiser were built (using HTS version 2.1 [1]), as an experimental TTS module for use in a speech-to-speech translation system to run on hand-held devices. The systems built are summarised in Table 1. For all systems built, the same training recipe was followed. It is essentially that described in [2]; the main differences include parameterisation and number of iterations of clustering. Acoustic parameters consisted of 25-dimensional vectors of mel cepstral coefficients, log F0, and first and second order derivatives of these. Instead of two iterations of clustering, 10 were performed; as shown in Section 3.1, this number of iterations is required before likelihood scores and model sizes converge.

The only difference between the 6 systems is in the contexts used to define context-dependent phones. Four different feature-sets are used: 2 conventional ones (basic positional information and part-of-speech), and 2 experimental ones (features taken from a language model, and automatically found word categories). These feature-sets are denoted Base, POS, LM and CAT in Table 1, respectively. In the context-clustering stages of training, each of the 6 systems used a question set that queries a different permutation of these four sets. As in [2], a Minimum Description Length criterion is used to determine model size; we used unweighted description length for all systems built.

The system denoted B in Table 1 is our baseline system, incorporating only positional information trivially extracted from lexicon and utterance text. System BP is our topline, and assumes the availability of a POS tagger; our aim is to add cheaply-obtained features in an effort to improve the performance of B so that it approaches or surpasses that of BP. To this end, baseline features are extended with language model features in system BL, with induced word categories in BC and with both of these in system BLC. System BLCP includes all of the sets of features, and was built to determine whether automatically found features might give performance gain when used on top of conventional POS features. The four sets of features will now be described in more detail.

2.1. Conventional Features

2.1.1. Basic Positional Questions

In addition to standard questions about a segment's phonetic characteristics and its immediate (quinphone) context, a set of basic features that are intended to capture prosodic features of speech is used in all systems, made up of features that can be straightforwardly extracted from the labelling of syllables and

phrases: position of phone in syllable, word, phrase and utterance; position of syllable in word phrase and utterance; position of word in phrase and utterance; and position of phrase in utterance. Features relating to lengths of units are also included: length of syllable in phones; length of word in phones and syllables; length of phrase in syllables and words; and length of utterance in syllables, words and phrases. Lexical stress of a syllable is also included as a feature. All these features are encoded via context-dependency at the phone level.

2.1.2. Part of Speech

System BP includes part of speech (POS) tags that are annotated in the database by a POS tagger. Features relating to POS were incorporated into System BP as follows: the POS of the word to which a phoneme belongs, and also the POS of the preceding and following words. As well as questions about single POS classes, we used linguistic knowledge to specify questions about sets of POS tags, such as "Is the {previous, current, following} word a {content word, type of pronoun, function word, *etc.*}?"

2.2. Experimental Features

2.2.1. Language Model Features

Motivation Systems BL and BLC incorporate features derived from a statistical (*n*-gram) language model of the type conventionally used in ASR and SMT. The decision to experiment with this type of feature was motivated by various facts. For example, function words tend to be high-frequency words; a simple unigram probability of a word might therefore be a good surrogate feature for knowledge-based features such as whether a word is function or content (see Section 2.1.2). Information Content is directly related to unigram probability, and has been shown to be a useful predictor of pitch accents [3]. It has been shown that frequency of occurrence has systematic effects on the way pitch accents are realised on accented syllables [4]. Higher-order *n*-grams encode the predictability of a word given a few words' context – this predictability has also been shown to be a useful predictor of pitch accents [5]. Importantly, *n*-gram models can make use of a resource – plain unannotated text – that, in comparison to either compiling rules for a POS tagger or manually annotating a database with POS for the training of such a tagger, is cheap to collect. What is more, the fact that TTS is happening in the context of of a speech-to-speech system means that the *n*-gram model and the features it provides are effectively 'free': the model has to be trained and stored for ASR and SMT components anyway, so using the features it provides involves no extra cost in terms of time or storage for TTS.

Method We used an existing LM that had already been trained for the ASR/SMT modules, on 27 million words of text; it is a 4-gram model using modified Kneser–Ney smoothing. Features were derived from it for TTS in the following way: word sequences were extracted from training corpus labels, 1-, 2-, and 3-gram probabilities were computed for each word in context. For use in decision-tree clustering, these probabilities were discretised in a very simple way: minimum and maximum values for probabilities associated with each length of *n*-gram were taken from the training data, and ranges for 30 evenly-spaced bins were calculated between these values. 30 was chosen arbitrarily as representing a quite fine level of quantization; coarser levels of quantization were represented by questions relating to ranges of bins. Context questions were added to systems BL, BLC and BLCP about bin-membership (and bin-range membership) of {previous, current, following} word. The same LM and discretisation bins were then used to assign corresponding features to the test set.

2.2.2. Induced category features

Motivation Parts of speech are most commonly discussed with reference to their syntactic, distributional properties. One way to view POS tags in conventional systems is as tags that capture some aspect of their word's distributional behaviour. One way to attempt to find classes that may act as surrogates for traditionally defined POS classes is to attempt discovery automatically based on words' distributional behaviour in a text corpus. Unsupervised methods for finding groups of words that behave in a distributionally similar way have proved useful for dealing with sparsity and improving performance in language modelling and bilingual alignment for SMT [6]. We use the method of [7], not least because an implementation of it is already used in the training of the system's SMT module.

Basically, the method seeks to find a class map that associates surface forms of words with some class label; the map is obtained in a way that seeks to maximize likelihood of the training data assuming a bigram model (class–class transition probabilities and class–word emission probabilities). For a predefined number of classes, the word–class map that maximizes likelihood of training data given this model is searched for.

The resulting classes will be dissimilar to POS classes in that they are not human-specified, and also that they are disjunct sets (not ambiguous as in the case of POS tags). The similarity to POS is that the classes are distributionally defined. Once again, the important thing about these features is that they are obtained in an unsupervised way, from unannotated text.

Method The class map was found using the same 27 million words as the LM had been trained on – the speech training data transcript is not included here, as omission of this text provides a simple and natural way to handle words that are not in the word–class map at run time. Words in the TTS training data but absent from the class map training data are replaced with a special class label for unseen words. One difficulty with the word-clustering method is that it provides no in-built way of determining suitable model complexity from the data: instead, the desired number of classes must be specified by the user. Instead of building systems and tuning this parameter (which would require time-consuming development cycles) we overcame this problem by including features that queried not only words' membership of classes, but also of sets of classes. Thus once again we allow a suitable level of granularity to emerge during acoustic model clustering. However, there is a further difficulty compared to the case of LM probabilities. The LM probability bins are based on numerical values that therefore have a natural ordering. The word classes, on the other hand, are represented by categorial values with no implicit ordering. To consider all possible binary partitions of a set of m categories would result in 2^{m-1} partitions, incurring prohibitive computational expense for sets of classes similar in size to POS tag-sets (40 gives over one billion 2-way partitions). We therefore employed the following method to induce a set of word classes that has sufficient structure to allow partitioning in a computationally efficient way, using repeated application of the method defined in [7]. We start by finding a class map with *n* classes. *n* is chosen large enough to represent a fine granularity of classes. In the present work, we set *n* as 100, bigger than most POS tagsets. After a class map has been found, the plain text used for training is rewritten using class symbols in place of words. *n* is then decremented by *d*, and a new mapping is found in the

Table 1: *Summary of Systems Built.*

System ID	Base	LM	CAT	POS
B	X			
BL	X	X		
BC	X		X	
BLC	X	X	X	
BP	X			X
BLCP	X	X	X	X

same way between 'words' (really class labels) and 'classes' (really sets of classes). We set d to 10 in the present work. The procedure is repeated until n is 1 or less. This produced a tree-structured set of word classes, where labels closer the root of the tree represent a coarser clustering of words. Context questions were added to systems BC, BLC, and BLCP querying the membership of the {previous, current, following} words in single word classes and also to the coarser sets of classes found in this way.

3. Analysis of Systems Built

3.1. Model Sizes and Fit to Training Data

Figures 1, 2, and 3 show model sizes for spectral envelope and logF0 components of the voices, and likelihood of training data given the models, respectively. It can be seen that approximately 10 iterations of clustering, re-estimation and untying are required for convergence of the models. Both Figures 2 and 3 suggest a natural clustering of the models into 2 groups: those including the CAT features – where the logF0 trees built are and which fit the training data more closely – and those which do not.

3.2. Systems' Usage of Context Features

We gathered statistics from the decision trees used to cluster acoustic models in all systems to have an idea to what extent the different types of features were used across the systems. As the new features we introduced we designed to improve the prosody of synthetic speech, we chose to analyse the trees built to cluster log F0 distributions, as we might expect changes in this part of the voices to reflect changes in the way prosody is modelled. To gather these statistics, we ran 50 utterances from the held-out set through the logF0 clustering trees of all 6 systems. A question is tallied each time a node in which it is asked is traversed. We group questions according to the groups as shown in Table 2; tallies are summed over these groups of questions and normalized by the total number of questions for each system (i.e. the columns of Table 2 represent percentages of questions asked within each system).

Table 2: *Systems' usage of phone context questions.*

System ID:	B	BL	BC	BLC	BP	BLCP
5-phones	66.9	64.8	60.3	60.8	62.4	59.0
Other basic	33.1	32.2	30.5	28.6	31.0	28.9
LM	-	3.0	-	2.1	-	2.0
CAT	-	-	9.1	8.5	-	6.3
POS	-	-	-	-	6.6	3.8

4. Objective Evaluation

As mentioned above, the labels and audio of 150 utterances from the corpus were held out from the training set. This enabled the computation of similarity measures between natural speech and speech synthesised from the corresponding labels. To be able to compare natural and synthesised parameters on a

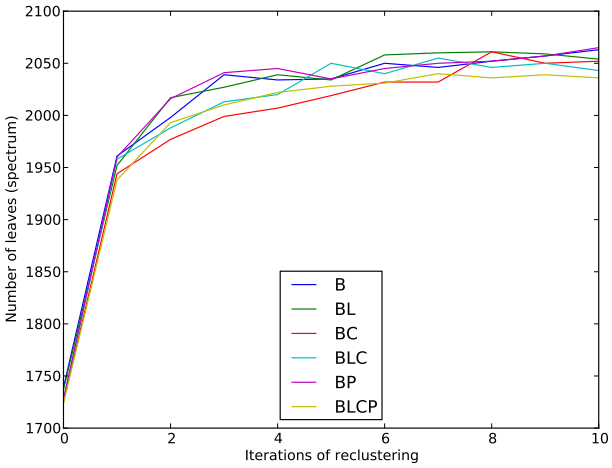

Figure 1: *Model size during voice-building: spectral envelope.*

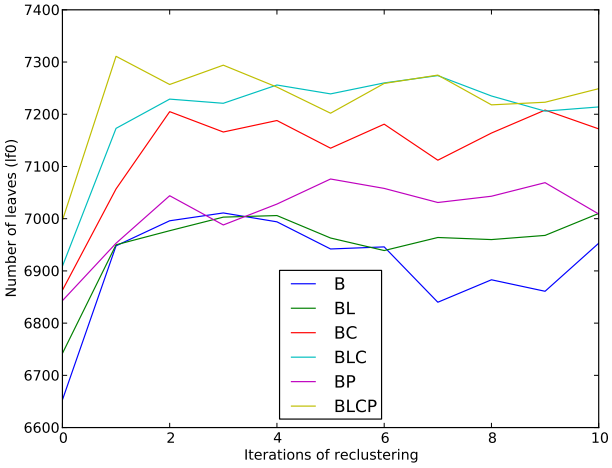

Figure 2: *Model size during voice-building: log F0.*

frame-by-frame basis, we constrained the synthesiser to use natural phone durations as annotated in the labels for the objective evaluation. Using this speech, we computed mean mel-cepstral distortion, voicing classification error, and RMSE of F0 (over correctly voiced frames) for each configuration of the synthesiser [8].

Results of the objective evaluation are given in Figure 4. It can be seen that inclusion of either the LM or CAT features in isolation produces only a slight difference in error with regard to the natural speech: a slight improvement for the source features and a slight worsening for the features representing spectral envelope. When used in conjunction, however, there is a marked improvement across all parts of the evaluation, especially in the log F0 part where system BLC obtains a better score than the topline system BP. Interestingly, when the unsupervised features are used on top of the topline features in system BLCP, there is a worsening of performance; this finding merits further investigation, and we suspect that it may be due to the greater number of features allowing the system to overfit the training data when context clustering trees are constructed.

5. Subjective Evaluation

As well as the objective measures computed as described above, a subjective evaluation of speech synthesised from the 150 held-out sentences (using speech segment durations as predicted by the models for this evaluation) was conducted as a listening test with human evaluators. Three of the systems built were

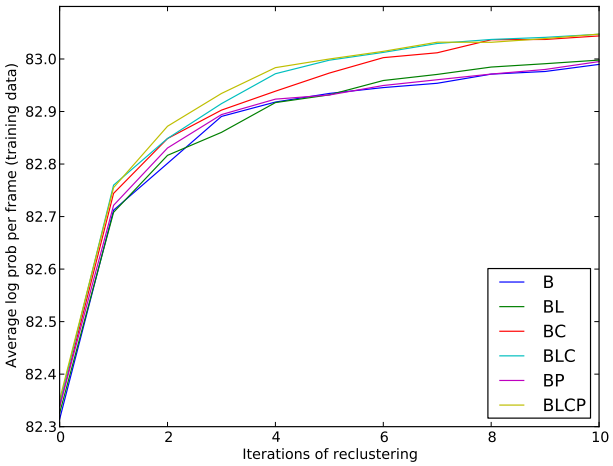

Figure 3: *Likelihood of training data given model during voice-building.*

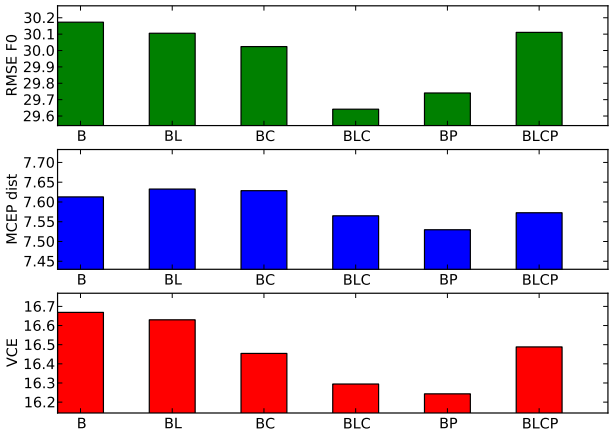

Figure 4: *Results of objective evaluation.*

included in this evaluation: baseline and topline systems, B and BP, and the experimental system that performed best in the objective evaluation, BLC.

The evaluation took the form of an AB test, in which comparison was made between the three systems in a pairwise fashion in terms of their perceived naturalness. 10 unpaid listeners each listened to 144 pairs of synthesised utterances. The text of the utterances was different for each of the 144, and each listener heard utterances from the same 144 texts. A third of the utterances (48) were assigned to each pair of systems being evaluated, and presentation order was balanced for each system pair. The assignment of utterance-texts to system-pairs was determined randomly for each listener; finally, the presentation order of sentence pairs was also randomised for each listener. Listeners were asked to choose the sentence that sounded more natural to them, and were also given the option of saying that neither sentence sounded more natural. The 'no preference' option was included because informal listening suggests that in many cases, utterances from the systems sound very similar.

Results of the subjective evaluation are shown in Figure 5. It can be seen that between 33 and 40 percent of sentences are rated 'no preference' in each comparison. Where a preference *is* rated, preferences are not clear-cut in either the B–BLC or B-BP comparisons: a binomial test (α=0.05), discounting 'no preference' choices, shows no significant difference from chance in either of these 2 cases. In the case of the BLC–BP comparison, the same test shows BLC to be preferred over BP significantly more often than chance when a preference for one of these two systems is recorded.

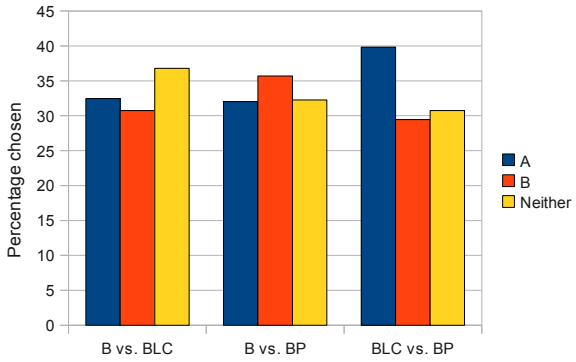

Figure 5: *Preference for Systems in Subjective Evaluation.*

6. Conclusions

The objective evaluations described here show that features acquired from text in an unsupervised manner enable the synthesis of acoustic parameters that are closer to natural reference speech than those generated by systems without access to such features. In some cases, the distance between generated parameters and reference ones when using these features is less even than in the case where conventional POS features are used. It is disappointing, however, that the picture presented by these objective evaluations is not supported by the majority of subjective preference tests. The objective scores, and the subjective preference for system BLC over BP, however, encourage us to think that the unsupervised extraction of features from text for TTS is a useful topic for on-going research, especially in situations where systems must be ported to languages where conventional linguistic resources are scarce or non-existent.

7. References

[1] H. Zen, T. Nose, J. Yamagishi, S. Sako, and K. Tokuda, "The HMM-based speech synthesis system (HTS) version 2.0," in *Proc. of Sixth ISCA Workshop on Speech Synthesis*, Aug. 2007, pp. 294–299.

[2] H. Zen, T. Toda, M. Nakamura, and K. Tokuda, "Details of Nitech HMM-based speech synthesis system for the Blizzard Challenge 2005," *IEICE Trans. Inf. & Syst.*, vol. E90-D, no. 1, pp. 325–333, Jan. 2007.

[3] S. Pan and K. R. McKeown, "Word informativeness and automatic pitch accent modeling," in *Proc. of joint SIGDAT conference on EMNLP and VLC*, 1999.

[4] K. Schweitzer, M. Walsh, B. Möbius, and H. Schütze, "Frequency of occurrence effects on pitch accent realisation," in *Proc. Interspeech*, Makuhari, Japan, Sep. 2010, pp. 138–141.

[5] S. Pan and J. Hirschberg, "Modeling local context for pitch accent prediction," in *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, 2000.

[6] R. Kneser and H. Ney, "Improved clustering techniques for class-based statistical language modelling," in *Proc. Eurospeech 1993*, Sep. 1993, pp. 973–976.

[7] F. J. Och, *Maximum-Likelihood-Schätzung von Wortkategorien mit Verfahren der kombinatorischen Optimierung, Studienarbeit, Universität Erlangen-Nürnberg, Germany*, 1995.

[8] K. Yu, B. Thomson, and S. Young, "From discontinuous to continuous f0 modelling in hmm-based speech synthesis," in *Proc. Speech Synthesis Workshop 2010*, Nara, Japan, Sep. 2010, pp. 94–99.