Festival *Multisyn* Voices for the 2007 Blizzard Challenge

Korin Richmond, Volker Strom, Robert Clark, Junichi Yamagishi and Sue Fitt

Centre for Speech Technology Research
University of Edinburgh, Edinburgh, United Kingdom
(korin|vstrom|robert|jyamagis|sue)@inf.ed.ac.uk

Abstract

This paper describes selected aspects of the Festival Multisyn entry to the Blizzard Challenge 2007. We provide an overview of the process of building the three required voices from the speech data provided. This paper focuses on new features of Multisyn which are currently under development and which have been employed in the system used for this Blizzard Challenge. These differences are the application of a more flexible phonetic lattice representation during forced alignment labelling and the use of a pitch accent target cost component. Finally, we also examine aspects of the speech data provided for this year's Blizzard Challenge and raise certain issues for discussion concerning the aim of comparing voices made with differing subsets of the data provided.

1. Introduction

`Multisyn` is a waveform synthesis module which has recently been added to the Festival speech synthesis system [1]. It provides a flexible, general implementation of unit selection and a set of associated voice building tools. Strong emphasis is placed on flexibility as a research tool on one hand, and a high level of automation using default settings during "standard" voice building on the other.

This paper accompanies the Festival Multisyn entry to the Blizzard Challenge 2007. Similar to the Blizzard Challenges of the previous two years ([2, 3]), the 2007 Blizzard Challenge required entrants to build three voices from the speech data provided by speaker "EM001", then submit a set of synthesised test sentences for evaluation. The first voice, labelled voice "A", used the entire voice database. Two smaller voices, "B" and "C" used subsections of the database. Voice "B" used the set of sentences from the ARCTIC database [4] which were recorded by the EM001 speaker. For voice "C", entrants were invited to perform their own text selection on the voice database prompts to select a subset of sentences no larger than the ARCTIC data set in terms of total duration of speech in seconds. Voices "B" and "C" are intended as a means to compare different text selection algorithms, as well as to evaluate the performance of synthesis systems when using more limited amounts of speech data.

Multisyn and the process of building voices for Multisyn is described in detail in [1]. In addition, entrants to the Blizzard Challenge this year have been asked to provide a separate system description in the form of a template questionnaire. For the reader's convenience this paper will provide a brief overview of Multisyn and the voices built. To limit redundancy, however, we will not repeat all details comprehensively. Instead, we aim to focus here on areas where the use of Multisyn differs from [1]. Those significant differences are two-fold. First, we will introduce a new technique we have been developing to help in forced alignment labelling. Next, we describe a target cost component which uses a simple pitch accent prediction model. Finally, we will discuss our experience of building voice "C", and highlight some issues we believe may complicate comparison of entrants' voices "B" and "C".

2. Multisyn voice building

We use our own Unisyn lexicon and phone set [5], so only used the prompts and associated wavefiles from the distributed data, performing all other processing for voice building from scratch. The first step of voice building involved some brief examination of the text prompts to find missing words and to add some of them to our lexicon, fix gross text normalisation problems and so on. Next, we used an automatic script to reduce the duration of any single silence found in a wavefile to a maximum of 50msec. From this point, the process for building Multisyn voices "A", "B" and "C" described in the remainder of this section was repeated separately for the relevant utterance subset for each voice.

We used HTK tools in a scripted process to perform forced alignment using frames of 12 MFCCs plus log energy (utterance based energy normalisation switched off) computed with a 10msec window and 2msec frame shift. The process began with single mixture monophone models with three emitting states, trained from a "flat start". Initial labelling used a single phone sequence predicted by the Festival Multisyn front end. However, as the process progressed with further iterations of reestimation, realignment, mixing up, adding a short pause tee model, and so on, we switched to using a phone lattice for alignment described in Section 3. Once labelling was completed, we used it to perform a waveform power factor normalisation of all waveforms in the database. This process looks at the energy in the vowels of each utterance to compute a single factor to scale its waveform. The power normalised waveforms were then used throughout the remainder of the voice building process, which began with repeating the whole labelling process.

Once the labelling had been completed, it was used to build utterance structures[1], which are used as part of the internal representation within a final Multisyn voice. At this stage, the text prompts were run through a simple pitch accent prediction model (see Section 4), and this information stored in the utterance structures. Additional information was also added to the utterance structures at this stage; for example, phones with a duration more than 2 standard deviations from the mean were flagged. Such information could be used later at unit selection time in the target cost function.

In addition to labelling and linguistic information stored in utterance files, Multisyn requires join cost coefficients and RELP synthesis parameters. To create the synthesis parameters, we first performed pitchmarking using a custom script which makes use of Entropic's `epochs`, `get_resid`, `get_f0` and `refcof` programs. We then used the `sig2fv` and `sigfilter` programs from the Edinburgh Speech Tools for lpc analysis and residual signal generation respectively. The

[1] a data structure defined in the Edinburgh Speech Tools library

Multisyn join cost uses three equally weighted components: spectral, f0 and log energy. The spectral and log energy join cost coefficients were taken from the MFCC files calculated by HTK's `HCopy` used for labelling. The f0 contours were provided by the ESPS program `get_f0`. All three of these feature streams were globally normalised and saved in the appropriate voice data structure.

During unit selection, Multisyn does not use any acoustic prosodic targets in terms of pitch or duration. Instead, the target cost is a weighted normalised sum of a series of components which consider the following: lexical stress, syllable position, word position, phrase position, part of speech, left and right phonetic context, "bad duration" and "bad f0". As mentioned above, "bad duration" is a flag which is set on a phone within a voice database utterance during voice building and suggests a segment should not be used. Similarly, the "bad f0" target cost component looks at a candidate unit's f0 at concatenation points, considering voicing status rather than a specific target f0 value. We have also used an additional target cost component for the presence or absence of a pitch accent on a vowel. This is described further in Section 4.

Finally, we stress that during concatenation of the best candidate unit sequence, Multisyn does not currently employ any signal processing apart from a simple overlap-add windowing at unit boundaries. No prosodic modification of candidate units is attempted and no spectral, amplitude or f0 interpolation is performed across concatenation boundaries.

3. Finite state phonetic lattice labelling

For all three voices for this Blizzard Challenge we employed a forced alignment system we have been developing which makes use of a finite state representation of the predicted phonetic realisation of the recorded prompts. The advantage of the finite state phonetic representation is that it makes it possible to elegantly encode and process a wide variety pronunciation variation during labelling of speech data. In the following two sections we first give a general introduction to how our phonetic lattice labelling works, and then give some more specific details of how the system was applied to building voices for this Blizzard Challenge.

3.1. General implementation

If we consider how forced alignment is standardly performed using HTK, for example, the user is required to provide, among other things, a pronunciation lexicon and word level transcription. The pronunciation lexicon contains a mapping between a given word and a corresponding sequence of phone model labels. During forced alignment, the HTK recognition engine loads the word level transcription and expands this into a recognition network, or "lattice", of phone models using the pronunciation dictionary. This lattice is then used to align against the sequence of acoustic parameter vectors. The predominant way to include pronunciation variation within this system is to use multiple entries in the lexicon for the same word. This approach generally suits speech recognition, but in the case of labelling for building a unit selection voice, we could perhaps profit from more flexibility. Complete flexibility is achieved if we compose the phone lattice directly and pass that to the recognition engine.

To build the phone lattice for a given prompt sentence, we first lookup each word in the lexicon and convert the phone string to a simple finite state structure. When a word is not found in the lexicon, we use the CART letter-to-sound rules the final festival voice would use to generate a phone string. Where multiple pronunciations for a word are found, we can combine these into a single finite state representation using the union operation. The finite state machines for the separate words are then concatenated in sequence to give a single representation of the sentence. The top finite state acceptor (FSA) in Figure 1 gives a simplified example of the result of this process for a phrase fragment "...wider economic...".

At this stage, there is little advantage over the standard HTK method, which would internally arrive at the same result. However, once we have a predicted phonetic realisation for a recording prompt in a finite state form, it is then straightforward to process this representation further in an elegant and robust way. This is useful to help perform simple tasks, such as splitting stops and affricates into separate symbols for their stop and release parts during forced alignment (done to identify a suitable concatenation point). More significantly, though, we can also robustly apply more complex context dependent postlexical rules, for example optional "r" epenthesis intervocalically across word boundaries for certain British English accents. This is indicated in the bottom FSA of Figure 1.

This may be conveniently achieved by writing rules in the form of context dependent regular expressions. It is then possible to automatically compile these rules into an equivalent finite state transducer which can operate on the input lattice which resulted from lexical lookup (e.g. top FSA in Figure 1). Several variations of compilation methods have been previously described to convert a system of handwritten context dependent mapping rules into an equivalent FST machine to perform the transduction, e.g. [6, 7, 8]. Note that the use of context dependent modifications is more flexible and powerful than the standard HTK methods. For example, a standard way to implement optional "r" epenthesis pronunciation variation using a pronunciation lexicon alone would be to include multiple entries for "wider", one of which contains the additional "r". However, this introduces a number of problems. The most significant problem is the absence of any mechanism to disallow "r" epenthesis in environments where a vowel does not follow.

The phonetic lattice alignment code has been implemented as a set of python modules which underlyingly use and extend the MIT Finite State Transducer Toolkit [9]. We use CSTR's Unisyn lexicon [5] to build voices and within the running synthesis system. For forced alignment, we use scripts which underlying make use of the HTK speech recognition library [10]. Finally, we are planning to make this labelling system publicly available once it reaches a more mature state of development.

3.2. Application to EM001 voice

Speaker EM001 exhibits a rather careful and deliberate approach to pronunciation during the recordings and uses a relatively slow rate of speech. This in fact tends to limit the applicability and usefulness of postlexical rules for the Blizzard Challenge voices somewhat. Postlexical rules are more usefully applied to the processes of more fluent and rapid connected speech. Thus, in building the three voices for the 2007 Blizzard Challenge, the sole postlexical rule we used was a "tap" rule. Under this rule, alveolar stops in an intervocalic cross word environment could undergo optional transformation to a tap. Specifically, the left phonetic context for this rule comprised the set of vowels together with /r, l, n/ (central and lateral approximants and alveolar nasal stop), while the right context contained just the set of vowels.

4. Pitch accent prediction

In this year's system, we have experimented with a simple pitch accent target cost function component. To use pitch accent prediction in the voices built for the Blizzard Challenge required three changes. First, we ran a pitch accent predictor on the text

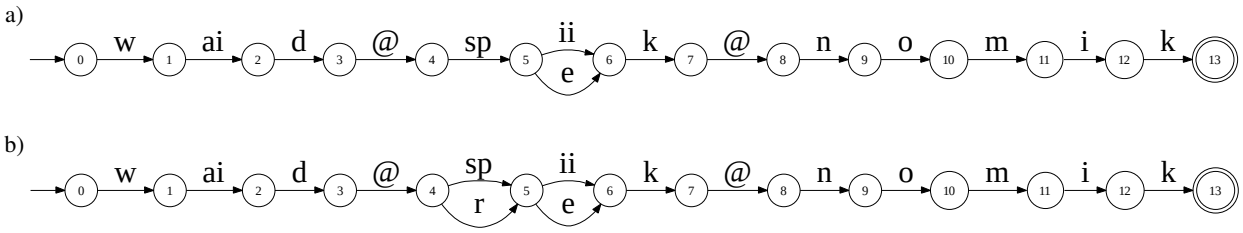

Figure 1: Toy example finite state phonetic lattices for the phrase fragment "wider economic": a) after lexical lookup, the lattice encodes multiple pronunciation variants for "economic" b) after additional "r" insertion postlexical rule, the input lattice (top) is modified to allow optional insertion of "r" (instead of short pause "sp").

prompts and flagged words with a predicted accent as such in the voice data structures. Next, at synthesis time, our front end linguistic processor was modified to run the accent predictor on the input sentence to be synthesised, and words with a predicted accent were similarly flagged. Finally, an additional target cost component compared the values of the pitch accent flag for the word associated with each target vowel and returned a suitable cost depending on whether they match or not.

The method for pitch accent prediction we used here is very simple. It is centred on a look-up table of probabilities that a word will be accented, or "accent ratios", along the lines of the approach described in [11]. The accent predictor simply looks up a word in this list. If the word is found and its probability for being accented is less than the threshold of 0.28, it is not accented. Otherwise it will receive an accent. These accent ratios are based on the BU Radio Corpus and six Switchboard dialogues. The list contains 157 words with an accent ratio of less than 0.28[2]. The pitch accent target cost component has recently been evaluated in a large scale listening test and was found to be beneficial [12].

5. Voice "C" and text selection

Entrants to the 2007 Blizzard Challenge were encouraged to enter a third voice with a voice database size equal to that of the ARCTIC subset, but with a freely selected subset of utterances. The purpose of this voice is to probe the performance of each team's text selection process, as well as to provide some insight into the suitability of the ARCTIC data set itself.

5.1. Text selection process

Ordinarily, when designing a prompt set for recording a unit selection voice database, we would seek to avoid longer sentences. They are generally harder to read, which means they are more taxing on the speaker and are more likely to slow down the recording process. In this case, however, since the sentences had been recorded already, we decided to relax this constraint.

In a simple greedy text selection process, sentences were chosen in an iterative way. First, the diphones present in the EM001 text prompts were subcategorised to include certain contextual features. The features we included were lexical stress, pitch accent and proximity to word boundary. Syllable boundary information was not used in the specification of diphone subtypes.

Next, sentences were ranked according to the number of context dependent diphones contained. The top ranking sentence was selected, then the ranking of the remaining sentences was recomputed to reflect the diphones now present in the subset of selected sentences. Sentences were selected one at a time in this way until the total time of the selected subset reached the prescribed threshold. This resulted in a subset comprising 431 utterances, with a total duration of 2908.75 seconds.

[2]using the accent ratio table in this way is essentially equivalent to using an (incomplete) list of English function words.

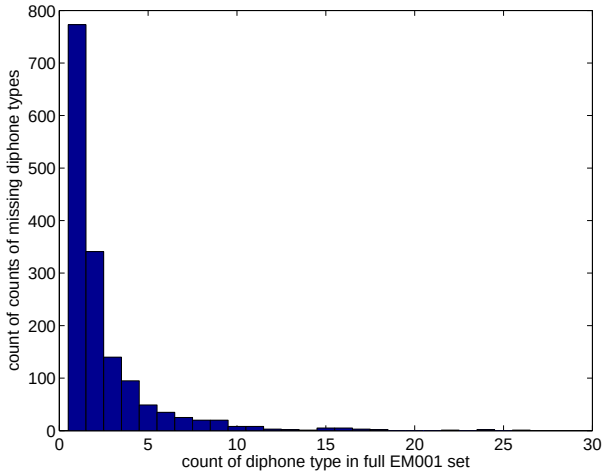

Figure 2: Histogram of counts of unique context dependent diphone types present in the full EM001 set which are missing from the selected subset used to build for voice "C".

Our definition of context dependent diphones implied a total of 6,199 distinct diphones with context in the entire EM001 corpus. Our selected subset for voice "C" contained 4,660 of these, which meant 1,539 were missing. Figure 2 shows a histogram of the missing diphone types in terms of their counts in the full EM001 data set. We see that the large majority of the missing diphone types only occur 1–5 times in the full EM001 dataset. For example, 773 of the diphone types which are missing from the selected subset only occur once in the full EM001 set, while only one diphone type which is missing occurred as many as 26 times in the full data set.

5.2. Evaluation problems

Although it is certainly interesting to compare different text selection algorithms against the ARCTIC sentence set, we suggest the way it has been performed this year could potentially confuse this comparison. The first issue to which we would like to draw attention concerns the consistency of the recorded speech material throughout the database. The second issue concerns the question of how far the full EM001 data set satisfies the selection criteria used by arbitrary text selection algorithms.

5.2.1. Consistency of recorded utterances

Figures 3–5 show plots of MFCC parameter means from the EM001 database taken in alphabetical file ordering. To produce

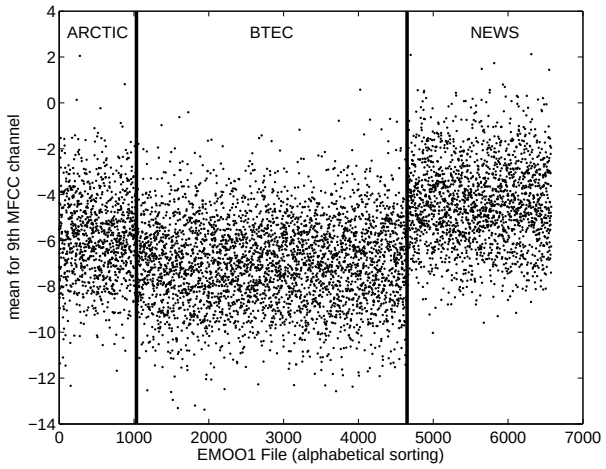

Figure 3: Mean value for 9th MFCC channel for each file of the EM001 voice database.

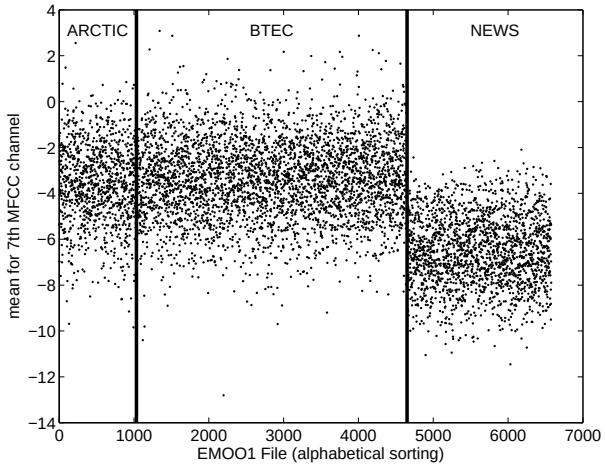

Figure 4: Mean value for 7th MFCC channel for each file of the EM001 voice database.

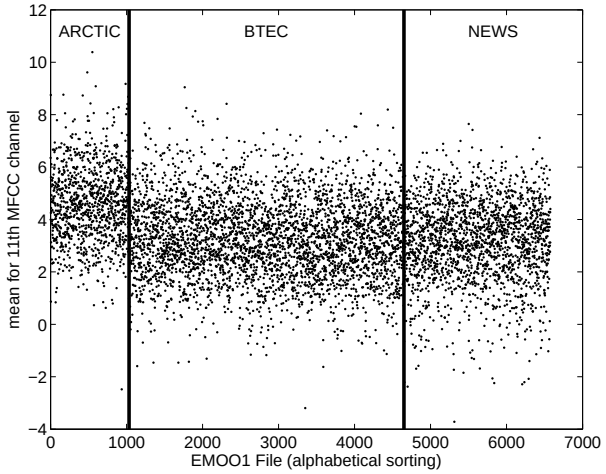

Figure 5: Mean value for 11th MFCC channel for each file of the EM001 voice database.

these plots we have taken all files in the EM001 data set in alphabetical ordering (along the x-axis) and calculated the mean MFCC parameters[3] for each file. In calculating these means, we have omitted the silence at the beginning and end of files using the labelling provided by the force alignment we conducted during voice building. A single selected dimension of this mean vector is then plotted in each of the Figures 3–5.

From these figures, we notice that there seem to be three distinct sections of the database, which correspond to the "ARCTIC", "BTEC" and "NEWS" file labels as indicated in the plots. Within each of these blocks, the MFCC mean varies randomly, but apparently uniformly so. Between these three sections, however, we observe marked differences. For example, compare the distributions of per-file means of the 9th (Fig. 3) and 7th (Fig. 4) MFCC parameters within the "NEWS" section with those from the other two sections of the database.

We naturally expect the MFCC means to vary "randomly" from file to file according to the phonetic content of the utterance contained. However, an obvious trend such as that exhibited in these plots suggests the influence of something more than phonetic variation alone. Specifically, we suspect this situation has arisen due to the significant difficulty of ensuring consistency throughout the many days necessary to record a speech corpus of this size. We have observed similar effects of inconsistency within other databases, both those we have recorded at CSTR, as well as other commercially recorded databases. Recording a speech corpus over time allows the introduction of variability, with potential sources ranging from the acoustic recording environment (e.g. microphone placement relative to speaker) to the quality of the speaker's own voice, which of course can vary over a very short space of time [13]. In addition, even the genre and nature of the prompts themselves can influence a speaker's reading style and voice characteristics.

Note that although we do not see any trends *within* each of the three sections of the EM001 data set, and that they appear relatively homogeneous, this does not imply that these subsections are free of the same variability and inconsistency. These plots have been produced by taking the files in alphabetical, and hence numerical, order. But it is not necessarily the case that the files were recorded in this order. In fact, it is likely the file ordering within the subsections has been randomised which has the effect of disguising inconsistency within the three sections. The inconsistency between the sections is evident purely because the genre identity tag has maintained three distinct groups.

Therefore, despite the probable randomisation of file order within sections, we infer from the patterns evident in Figures 3–5 that the speech data corresponding to the ARCTIC prompt set was recorded all together, and constitutes a reasonably consistent "block" of data. Meanwhile, the rest of the data seems to have been recorded at different times. This introduces inconsistency throughout the database, which a selection algorithm based entirely upon text features will not take account of. This means that unless it is explicitly and effectively dealt with by the synthesis system which uses the voice data, both at voice building time (e.g. using cepstral mean normalisation during forced alignment) and at synthesis time, voice "C" stands a high chance of being disadvantaged by selecting data indiscriminately from inconsistent subsections of the database. The forced alignment labelling may suffer because of the increased variance of the speech data. Unit selection may suffer because the spectral component of the join cost may result in a nonuniform probability of making joins across sections of the database, compared with the those joins within a single section. This has the effect of "partitioning" the voice database.

[3] extracted using HTK's HCopy as part of our force alignment processing, and also subsequently used in the Multisyn join cost

The Multisyn voice building process currently takes account of amplitude inconsistency, and attempts waveform power normalisation on a per-utterance basis. However, other sources of inconsistency, most notably spectral inconsistency are not currently addressed. This means that Multisyn voice "C" is potentially affected by database inconsistency, which introduces uncertainty and confusion in any comparison between voices "B" and "C". Within the subset of 431 sentences we selected to build voice "C", 261 came from the "NEWS" section, 169 came from the "BTEC" section, and the remaining 36 came from the "ARCTIC" section.

This issue of inconsistency can potentially affect the comparison between the "C" voices from different entrants. For example, according to our automatic phonetic transcriptions of the EM001 sentence set, the minimum number of phones contained in a single sentence within the "NEWS" section is 52. Meanwhile, the "BTEC" section contains 1,374 sentences with less than 52 phones. Although we have not done so here, it is not unreasonable for a text selection strategy to favour short sentences, in which case a large majority may be selected from the "BTEC" section. This would result in avoiding the large discontinuity we observe in Figures 3 and 4 and could potentially confer an advantage which is in fact unrelated to the text selection algorithm per se.

The problem has the potential, however, to introduce most confusion into the comparison between entrants' voices "B" and "C", as there is most likely to be a bias in favour of the ARCTIC subset, which seems to have been recorded as a single block. We suggest there are at least two ways of avoiding this bias in future challenges. One way would be to provide a database without the inconsistency we observe here, for example through post-processing. This is likely to be rather difficult to realise, and our own previous attempts have failed to find a satisfactory solution, although [14] reported some success. A second, simpler way would be to record the set of ARCTIC sentences randomly throughout the recording of a future Blizzard Challenge corpus.

5.2.2. Selection criteria coverage

The second problem inherent in attempting to compare text selection processes in this way arises from differing selection criteria. It is usual to choose text selection criteria (i.e. which diphone context features to consider) which complement the synthesis system's target cost function. Hence the criteria may vary between systems.

The set of ARCTIC sentences was selected from a very large amount of text, and so the possibility for the algorithm to reach its optimal subset in terms of the selection criteria it used is maximised. In contrast, the text selection required for voice "C" was performed on a far smaller set of sentences. Although, admittedly, it is likely to be phonetically much richer than if the same number of sentences had been selected randomly from a large corpus, it is possible that the initial set of sentences does not contain a sufficient variety of material to satisfy the selection criteria of arbitrary text selection systems. This again may tend to accord an inherent advantage to voice "B".

6. Conclusion

We have introduced two new features of the Multisyn unit selection system. We have also raised issues for discussion concerning the comparison of voices built with differing subsets of the provided data. Finally, we note that, as in previous years, participating in this Blizzard Challenge has proved both interesting and useful.

7. Acknowledgments

Korin Richmond is currently supported by EPSRC grant EP/E027741/1. Many thanks to Lee Hetherington for making the MITFST toolkit available under a BSD-style license, and for other technical guidance. Thanks to A.Nenkova for processing the Blizzard text prompts for pitch accent prediction.

8. References

[1] R. A. J. Clark, K. Richmond, and S. King, "Multisyn: Open-domain unit selection for the Festival speech synthesis system," *Speech Communication*, vol. 49, no. 4, pp. 317–330, 2007.

[2] R. Clark, K. Richmond, V. Strom, and S. King, "Multisyn voice for the Blizzard Challenge 2006," in *Proc. Blizzard Challenge Workshop (Interspeech Satellite)*, Pittsburgh, USA, Sept. 2006, (http://festvox.org/blizzard/blizzard2006.html).

[3] R. A. Clark, K. Richmond, and S. King, "Multisyn voices from ARCTIC data for the Blizzard challenge," in *Proc. Interspeech 2005*, Sept. 2005.

[4] J. Kominek and A. Black, "The CMU ARCTIC speech databases," in *5th ISCA Speech Synthesis Workshop*, Pittsburgh, PA, 2004, pp. 223–224.

[5] S. Fitt and S. Isard, "Synthesis of regional English using a keyword lexicon," in *Proc. Eurospeech '99*, vol. 2, Budapest, 1999, pp. 823–826.

[6] M. Mohri and R. Sproat, "An efficient compiler for weighted rewrite rules," in *Proc. 34th annual meeting of Association for Computational Linguistics*, 1996, pp. 231–238.

[7] R. Kaplan and M. Kay, "Regular models of phonological rule systems," *Computational Linguistics*, vol. 20, no. 3, pp. 331–378, Sep 1994.

[8] L. Karttunen, "The replace operator," in *Proc. 33th annual meeting of Association for Computational Linguistics*, 1995, pp. 16–23.

[9] L. Hetherington, "The MIT finite-state transducer toolkit for speech and language processing," in *Proc. ICSLP*, 2004.

[10] S. Young, G. Evermann, D. Kershaw, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev, and P. Woodland, *The HTK Book (for HTK version 3.2)*, Cambridge University Engineering Department, 2002.

[11] J. Brenier, A. Nenkova, A. Kothari, L. Whitton, D. Beaver, and D. Jurafsky, "The (non)utility of linguistic features for predicting prominence on spontaneous speech," in *IEEE/ACL 2006 Workshop on Spoken Language Technology*, 2006.

[12] V. Strom, A. Nenkova, R. Clark, Y. Vazquez-Alvarez, J. Brenier, S. King, and D. Jurafsky, "Modelling prominence and emphasis improves unit-selection synthesis," in *Proc. Interspeech*, Antwerp, 2007.

[13] H. Kawai and M. Tsuzaki, "Study on time-dependent voice quality variation in a large-scale single speaker speech corpus used for speech synthesis," in *Proc. IEEE Workshop on Speech Synthesis*, 2002, pp. 15–18.

[14] Y. Stylianou, "Assessment and correction of voice quality variabilities in large speech databases for concatentative speech synthesis," in *Proc. ICASSP-99*, Phoenix, Arizona, Mar. 1999, pp. 377–380.