The Lessac Technologies System for Blizzard Challenge 2011

Reiner Wilhelms-Tricarico, Brian Mottershead, Rattima Nitisaroj, Michael Baumgartner, John Reichenbach, Gary Marple

Lessac Technologies, Inc., USA

{reiner.wilhelms, brian.mottershead, rattima.nitisaroj, mike.baumgartner, john.reichenbach, gary.marple} @lessactech.com

Abstract

Lessac Technologies has developed a technology for concatenative speech synthesis based on a novel approach for describing speech in which expressivity, voice quality, and speaking style are fundamental. The main aspect of our system is that instead of traditional phonetic symbols, we use a much more fine-grained and richer set of entities called Lessemes to describe speech and to label units, which allow a richer and more precise characterization of speech sounds. The front-end part of our synthesizer translates plain input text into a sequence of these units by syntactic parsing and applying a set of rules developed from expertise. We use a Bayesian method to obtain a particular trainable mapping from linguistic and prosodic features encoded in the Lessemes to a trajectory in the acoustic parameter space. Unit selection consists of selecting the best candidate units from a data base to match them to the target trajectory, while minimizing discontinuities between them.

Index Terms: speech synthesis, Blizzard Challenge, Lesseme.

1. Introduction

This is our second entry to the Blizzard Challenge. For 2011 , Lessac Technologies provided the "Nancy" data base of recordings and associated data to the community as the basis of the 2011 Challenge. The "Nancy" voice corpus consists of 16 hours of high quality recordings of natural expressive human speech made in an anechoic chamber at a 96K sampling rate during 2007 and 2008.

One of our intentions in making this "Nancy" voice corpus available to the research community, was to find out to what extent our approach to speech synthesis has advantages over others, and to confirm that the advantages are not just attributable to our approach or a particular voice model. For the Blizzard submission we did not specifically build a new model; we used our standard approach to synthesize the test data.

Section 2 provides a description of our text-to-speech system. Section 3 explains the Lessac process of building the "Nancy" voice. In the first half of Section 3, we explain the approach we used in developing the pre-cursor elements of building a synthesizer, such as prompts, pitch-marks, and phonetic labels. These elements of the "Nancy" voice corpus were made available to each Blizzard Challenge 2011 entrant. In the second half of Section 3, we outline how we used the data available in the "Nancy" voice corpus to build our complete Lessac text-to-speech synthesis system. Results from the listening test and related discussion can be found in Section 4.

2. Lessac Technologies Text-to-Speech System

Similar to other systems, Lessac Technologies text-to-speech system consists of two main components: the front-end, which takes plain text as input and outputs a sequence of graphic symbols, and the back-end, which takes the graphic symbols as input to produce synthesized speech as output. In what follows, we briefly discuss the properties that distinguish our system from others and, we believe, play an important role in producing expressive synthesized speech.

2.1. Use of Lessemes

Successful production of natural sounding synthesized speech requires developing a sufficiently accurate symbolic set of sound representations that can be derived from the input text, and that relate the input text to be pronounced with the corresponding synthesized speech utterances that are heard by the listener. Rather than adopting traditional symbolic representations, such as IPA, SAMPA, or ARPAbet, Lessac Technologies has derived an extended set of symbolic representations called Lessemes from the phonosensory symbol set for expressive speech as conceived by Arthur Lessac [1]. The Lesseme system for annotating text explicitly captures the musicality of speech, and from the start avoids the artificial separation of prosodic and linguistic features of speech.

In their basic form and meaning, Lessemes are symbolic representations that carry in their base form segmental information just like traditional symbolic representations. To be able to describe speech more accurately and to include in the symbol set information that is not carried by a typical phonetic symbol, each base Lesseme can be sub-typed into several more specific symbols which then represent phonetic information found in traditional phonetic symbols plus descriptors for co-articulation and suprasegmental information. Acoustic data demonstrate different properties of a set of Lessemes which are normally collapsed under one phonetic label in other systems [2].

For General American English, with the present Lesseme specification, there can be as many as 1,500 different Lessemes. Compared to other sets of representations which usually contain about 50 symbols, Lessemes allow more fine-grained distinction of sounds. Units of the same type share closely similar acoustic properties. By having suprasegmental information directly encoded in Lessemes, we believe our system can target available units for concatenation better than a system with a relatively impoverished intonation annotation scheme. This should be useful especially when trying to produce expressive speech from a very large database.

2.2. Front-end with extensive linguistic knowledge

The front-end which derives Lessemes from plain text input is a rules-based system. The rules are based on expert linguistic knowledge from a wide variety of fields including phonetics, phonology, morphology, syntax, light semantics, and discourse. Simplistically, the Lessac front-end labels text, building from, at the lowest level, letters, spaces and punctuation marks. These letters, spaces and punctuation marks are interpreted by the front-end, and assembled as syllables, words, phrases, sentences, and paragraphs to be spoken, along with context-aware labeling for appropriate co-articulations, intonation, inflection, and prosodic breaks.

First, the input text is processed by a syntactic parser which generates the most likely syntactic tree for each sentence, and tags words with part-of-speech (POS) information. In the next step, words are transcribed by use of a pronunciation dictionary into base Lessemes accompanied by lexical stress. Homograph disambiguation based on POS tags takes place at this step. Subsequent processing steps modify the base Lessemes by making successive decisions based on the overall phrase and sentence structure. In particular, prosodic breaks are inserted in meaningful places by taking into consideration factors such as punctuation, phrase length, syntactic constituency, and balance. In most phrases, an operative word is marked which carries the highest pitch prominence within the phrase. In addition, Lessemes are assigned inflection profiles and one of two degrees of emphasis. Context-based co-articulations across word boundaries are also captured. The result is a full Lesseme for each sound which encodes expressive intonational content in addition to segmental information found in traditional phonetic symbols.

The front-end process is able to develop a complete Lesseme label stream with plain normally punctuated text as the sole input. This Lesseme stream is delivered to the signal processing back-end.

Lessac made the output of this front-end process available to other entrants.

2.3. Voice database construction

In addition to the machine readable form used as the input to the signal processing back-end, Lessemes are also used in creating new voices, namely to automatically generate a human readable graphic output stream which can be thought of as annotated text plus a musical score, as illustrated in figure 1.

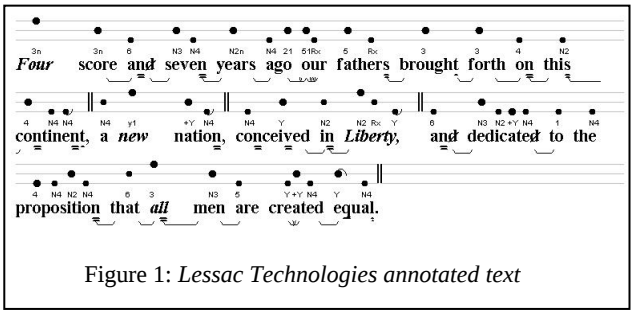


Figure 1: *Lessac Technologies annotated text*

In the annotation, vowel orthographic forms are designated with Arthur Lessec's phonosensory symbols. Consonant orthographic forms are marked with information indicating whether the consonant is sustainable (double underlined) or percussive, i.e. pronounced with a brief contact within the mouth (single underlined), as well as how the consonant is linked to the next sound in connected speech. The musical score on top of the orthographic forms depicts notes which represents the intonation pattern that a person with sufficient voice training can follow. Each syllable corresponds to a note. Higher notes are pronounced with higher pitch. Large notes define stressed syllables while small notes refer to unstressed syllables. Some notes are further specified with an inflection, which reflects a particular shape of pitch movement within the syllable.

During the voice database construction, the text to-be-recorded is first processed by the front-end, yielding the stream of Lessemes. The resulting stream is then transformed into a human readable form, as seen in figure 1, which we use as the combined script and score for the trained voice talent during the recordings. The way the voice talent records the prompts is controlled by the annotated text and musical score. The recordings of the prompts are then segmented and labeled with the same Lessemes that underlie the script and score that the voice talent followed. The fact that the same Lessemes are output for the voice talent script as well as the labeling of the database creates a direct link between each speech snippet and its Lesseme label, thus a high degree of correspondence between the symbols and the sounds as actually recorded by the voice talent. Such high degree of symbol-to-sound correspondence is not guaranteed in the typical voice database construction, where the voice talent sees only plain text and the subsequent recordings are labeled with the symbols generated by the front-end. We make use of this correspondence in the unit selection process by evaluating units in the data base according to the context dependent linguistic and prosodic features, in order to preselect a subset of unit candidates, which are then evaluated by the model described in the following.

2.4. Hierarchical Mixture of Experts for mapping linguistic features to acoustic parameters

To enhance methods for target cost calculation and unit selection, we apply the Hierarchical Mixture of Experts (HME) model [3] [4] to learn the parameters of a statistical model of the relationship between the Lesseme representation of the input text and the ideal acoustic observables in the recordings.

A functional diagram of the HME model is shown in figure 2.

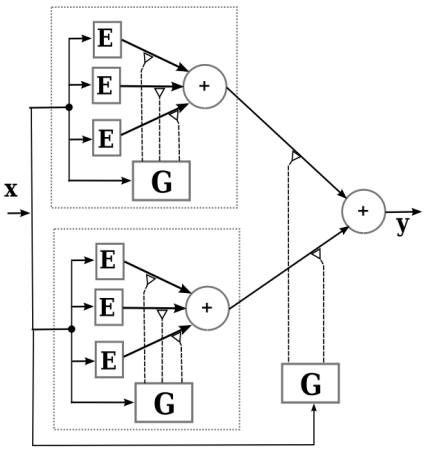


Figure 2: *Hierarchical Mixture of Experts model. (E: experts, G: gates, x: input, y: output)*

The HME model applied to the problem of mapping prosodic features to acoustic observables makes use of the interpretation of the model as a parameterized mixture of Gaussians. Each expert in the model represents one multi-dimensional normal distribution with a variable expectation vector that depends on the input x. The parameters for each expert also include a full covariance matrix that is estimated and updated during the training. Each block of experts in a

group or clique (Figure 2 shows 3 experts in each of 2 cliques) together with a gating network represent one mixture of Gaussians whereby the mixture coefficients are computed in the gates as a function of the input. Multiple groups of experts can be combined by another gate in a similar way. The complete network represents a mixture of Gaussians whose parameters are trained from pairs of known input and output. During the learning process, the parameters in the experts and gates are adjusted so that, for a given known input x, the probability of obtaining the desired known output y is maximized over all available data.

In our application of the HME model, the input x includes the linguistic and prosodic features and the output y are acoustic observables, which include MFCC's, F0, duration, and intensity, mostly the same type of parameters used in database segmentation, see below. The model is applied and trained as a recurrent system, which means that the predictions of acoustic observables, y[n], for one sound at time index n are included in the input x[n+1] for the prediction of the next y[n+1].

We use supervised learning with the HME model to map linguistic feature sequences to a trajectory in the acoustic parameter space, which is represented by via points and for some of the parameters their velocity or rate of change. The structure of the model is shown in figure 3. The system steps through a sequence of Lessemes and predicts for each Lesseme the vector of acoustic parameters that specify the unit, whereby the input to the model consists of the feature information of the previous, the current and the next two Lessemes. Further, by feeding back the previously predicted acoustic parameter vectors as input to the model, the model becomes partially auto-regressive. This facilitates the learning task because the model only has to learn to predict the current acoustic vector conditioned on the last two acoustic vectors and the input linguistic features. Learning proceeds in two phases. Initially, the looped-back input to the model consists of the actual acoustic vectors until the model begins to converge. Then, training is continued by having the predictions for the last two time slots become inputs for the prediction of the current time slot. Learning then proceeds by repeatedly processing a large number of sentences in the database, until the error variance can not be lowered further.

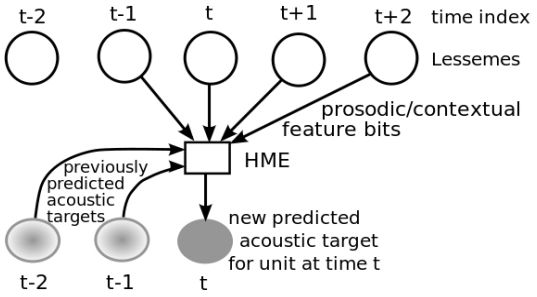


Figure 3: *Recurrent and partially auto-regressive prediction of intonation contour and other acoustic targets by HME*

During the target cost calculation process, we compute the cost as the distance of the acoustic parameters of a candidate unit from the ideal trajectory, which is in turn directly predicted from the linguistic feature variables. This distance measure makes use of the predicted mixture covariance matrix which is obtained by combining the experts' covariances according to the gating weights, see Figure 2. To reduce processing time, we reduce the number of candidates first by applying a rapid search with binary patterns generated from some of the features, and then compute the exact target cost for a smaller subset of close candidates. Since the HME provides the parameters of a probability density in the acoustic parameter space, we compute for the remaining candidates their probability under this distribution and use as target cost a penalty that is proportional to the negative logarithm of the candidates' probability.

Using the Lesseme representation of speech sounds, the output of the front-end results in a large number of features, which is augmented further by bundling neighboring features as shown in the figure 3. The HME model overcomes the sparsity problem in the data base by mapping the Lesseme features and context onto the acoustic parameter space as a target trajectory. At the same time it automatically provides a variable metric near the target trajectory, against which the candidates in the data base are matched during unit-selection.

3. Building 'Nancy' Voice

For the Blizzard Challenge 2011 we did not need to build a new voice, since we provided our already existing voice database to all participants. The following describes the steps that were taken earlier to create this voice.

3.1. Transcription to Lessemes

The speaker, Nancy Krebs, is a professional voice teacher and instructor for voice acting with the Lessac Institute. She was actively involved in the methodological layout and design for the symbolic system later developed by Lessac Technologies, which is closely related to the pictorial method of speech annotation shown in an example in Figure 1. Lessac Technologies then developed a computational method that allows us to generate automatically from arbitrary text the sequence of Lessemes that can then be presented in a form as shown in Figure 1 to the voice actor, while at the same time it provides the input information for the synthesizer's back-end.

For the segmentation of the original recordings in order to create the voice data base, the large number of Lessemes is usually a disadvantage because the number of possible states in an HMM based segmentation, as used in the festvox toolbox, is much larger than for a system based on a traditional phoneme set. To circumvent this sparsity problem we made use of the hierarchical organization of the Lessemes; each Lesseme label can be fully or partially collapsed into a much smaller number of less fully described Lesseme labels, with base Lessemes, similar to phonemes, being the lowest level. We can then train an HMM model using this collapsed inventory of symbols, and later refine both the HMM model and the segmentation by including more information into the HMM models.

Regarding the dictionary, we used an American English pronunciation dictionary to transcribe the words into Lessemes.

The acoustic features used for segmentation and similarly for the training of the HME model, were 12 MCEPS coefficients and their rates of change, together with F0, intensity, and zero crossing rates, but reduced to a lower dimensional representation by principal component analysis. The EHMM model in speech tools was used, with some minor modifications for processing the segmentation.

3.2. Pitch-Marking

Since it was the weakest point of our technology as presented in the last Blizzard Challenge, we put a significant portion of our recent effort into better pitch-marking and concatenation methods. We have observed that minimal pitch mismatches can cause noticeable synthesis artifacts. These artifacts can often be minimized by adjusting pitch around the join point to meet at the mean pitch of the ends of the units to be joined.

Herein is a dilemma: If the specific pitch marks are not accurate, then we can assume the adjustment of pitch will also be inaccurate. Experimentation has found that pitch synchronous techniques using a more robust pitch marking

technique yield very satisfactory results and reduce synthesis artifacts.

Instead of using pitch marks generated by Praat we are now using techniques developed by Mike Baumgartner, one of the authors. The system we used for Blizzard 2011 is a parameter driven system which relies on an expert adjusting the parameters to obtain the statistically best performance for a given voice talent. It relies on analyzing the signal to place pitch markers within specifiable probability boundaries of the most likely place of glottal closure. This is also a requirement for our new concatenation method which is implemented as a completely separate module and can in principle run on a separate server.

The pitch-marking program uses two checks to determine pitch. The first is a modified cross correlation. This cross correlation is performed several times with stepped window sizes that are limited to the pitch range of the voice talent. The best cross correlation performance obtained provides pitch prediction by the window size of the best performance. Another robust technique of pitch determination is the product of (vectors) of the frequency bins of the DFT of a windowed speech segment. The bins are multiplied cumulatively in increments of n = 1, 2, 3, etc. Only half of the DFT frequency bins are used, what is left over is zero padded. If the speech signal is periodic, a nice peak corresponding to the first harmonic appears in the cumulative product, the peak is formed by the product of energy of the 1^{st}, 2^{nd}, 3^{rd} harmonic etc. The performance of these two techniques tend to complement each other. When results are not in agreement, this is a good indication of unvoiced segments.

Once the pitch is determined, then the glottal closure instant is estimated. The current estimation is an elementary one. The speech signal has a higher slew rate after the glottal closure instant. A moving window is used for a segment of speech. Using the pitch of the signal to determine window size, the window is used in halves. The sum of the differences in the samples in the right half is divided by the sum of the differences in the left half. This gives a rough peak and a starting point for finding glottal closures. The speech signal is window averaged to low pass the signal (window size is one of the parameters). This removes the higher formant frequencies that would give several zero crossings.

We are currently completing the work for a fully automatic pitch-marking method by an innovative use of neural network machine learning techniques to successively calculate and weight pitch-marking parameters at several levels of abstraction. This has proven to further improve pitch marking, and will be used for our Blizzard 2012 demonstration.

3.3. Database creation

As our labeling and metrics for prosodic structure are different from methods presently used; we modified Festival feature functions to produce relevant linguistic features at segment, syllable, word, and phrase levels based on the Lessemes and prosodic breaks that the front-end provides as output. The end time of each unit came from the label files produced by automatic segmentation. Our segmentation procedures are based on a slightly modified version of the EHMM software that is part of speech tools and Festvox. Acoustic parameters were computed for each prompt, and a dimensionality reduction was obtained by principal component analysis. The resulting set of parameters were then used in building the HMM model for segmentation. For building the catalogue, all the linguistic features coming from the front-end analysis and the acoustic parameters were collected into a binary catalog file, which was then used to train the HME model off-line. The same binary catalogue is called by the synthesizer during run-time.

3.4. Synthesizer

While Lessemes help narrow the pool of candidates for unit selection and enable more precise targeting, labeling units with Lessemes can lead to the problem of non-existing or a sparse number of units of particular labels in the database, especially in a small database. We handle this problem by incorporating a set of fail-over rules. Whenever the target Lesseme has a very limited number of or even no matching candidates in the database, the fail-over rules look for closely matched Lessemes, e.g., those with a different inflection or pitch level, to include among the candidates for the target and join cost calculations. The target cost is computed as a weighted distance to the acoustic target trajectory that is generated by the HME model. The target penalty cost is derived from a logarithmic probability that can be computed for each candidate using the parameters provided by the HME model, namely target acoustic feature vector and covariances.

Similar to Kominek [5], our join cost calculation discourages joins between sonorant sounds. The join penalty varies depending on the types of sonorants being joined. For example, the join between two vowels gets a higher penalty than the join between a vowel and an onset lateral sound. We also make use of the HME output, namely the variance information, to modulate the spectral weights used in the join cost computation.

3.4 Concatenation

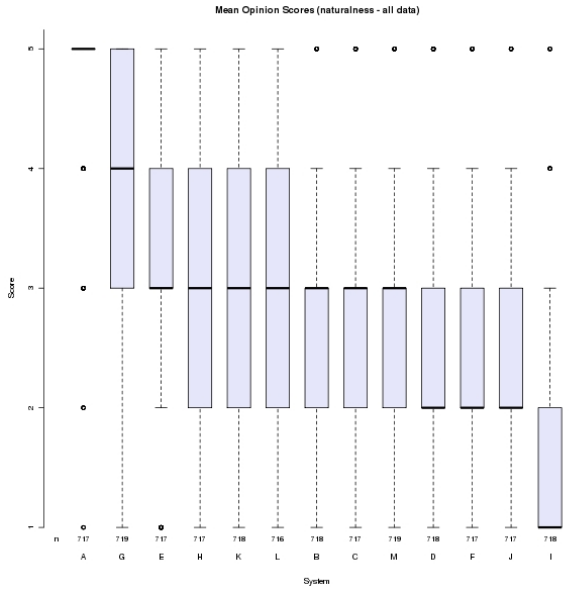
After the best units are selected, they are concatenated together in a process that works entirely in the time-domain. For this we are no longer using Festival but instead built a separate module which receives only the information about the units to be concatenated. The concatenation of voiced sounds is done pitch synchronously, and some mutual adjustments of two sounds that are concatenated are made to increase the coherence and to reduce clicks and warbles. F0 modifications and also duration modifications are also done independently of Festival in the concatenation module, using information that is transmitted to the concatenation module from the HME model.

4. Results and Discussion

Twelve systems participated in the EH1 task (building a voice from the full dataset). In addition, the original speaker's voice was evaluated as a benchmark, or pseudo thirteenth system (system A). During the online evaluation of the task, listeners were asked (i) to judge how similar a system is to the original speaker, (ii) to provide mean opinion scores (MOS) representing how natural or unnatural the utterances from the news and novel domains sound, (iii) to transcribe addresses read by the synthesizer, and (iv) to listen to synthesized semantically unpredictable sentences (SUS) and transcribe what they heard. The listeners included paid participants, volunteers, speech experts, native and non-native English speakers. Results for our system in comparison with standard Festival unit-selection systems and others are presented below.

4.1. Naturalness and similarity to original speaker

A 5-point mean opinion scale (MOS) was used to evaluate both how natural synthesized speech sounds, and how similar synthesized speech sounds to the original voice. With respect to naturalness of our synthesized speech, Lessac Technologies (system E) received a mean MOS score of 3.3 for all data and a median of 3.



For similarity to the original speaker, we received a mean score of 3.1 and median of 3. Overall we were ranked in second place, based on pairwise Wilcoxon signed rank tests. One system (system G) ranked higher on a statistically significant basis than our system and all others.

4.2. Word error rates

The increased expressiveness of the "Nancy" database in 2011 vs. the "RJS" databases in 2010 led to higher overall word error rates for most systems, as exemplified by the lower word error rate accuracy of the natural voice, that is, the original speaker reading "nonsense" sentences. The mean error rate for the natural voice (system A) was 17% this year versus 12% in last year's SUS test.

Despite the increased difficulty, our word error rate improved. For reading addresses, our median word error rate was 10.5% and the mean error rate was 15%. For the semantically unpredictable sentences (SUS) we received a median error rate of 14% and a mean rate of 22%. Overall, our median word error rate was 12.5%, and the median error rate was 19%. All of these scores are a significant improvement over our system's performance last year.

The Wilcoxon signed rank test resulted in little information that would give a significant rank ordering of the different systems. Based on the Wilcoxon signed rank test, our word error rate is worse than natural recorded speech, and comparable to the other systems (worse than none, and better than only one to a statistically significant degree). In other words, for nonsense sentences our system gets by with similar word recognition rates as most other systems.

5. Conclusions

We have made good progress in producing near natural sounding synthesized human speech highly similar to the original speaker. We attribute much of this progress to our recent improvements in the signal processing used for concatenation, which we indicated as our weakest point after last year's competition.

The overall performance of our system as one of the best in the Blizzard Challenge (2nd, closely followed by another system) gives us some confidence in support of our general strategy to try to represent and capture in the synthesis model idiosyncratic properties of the original voice that are not directly represented by known explicit models. For the symbolic representation of speech sounds for synthesis we use a novel method that is a departure from traditional phonetics by introducing Lessemes, which carry both segmental and suprasegmental information and allow for much more fine grained tagging of speech. This tagging process is done fully automatically, starting from plain text. Accordingly, the processing done by the front-end results in a very rich stream of features that are encoded with the speech samples in the database. We use methods of machine learning to create a sufficiently comprehensive model of the voice without having to make too many assumptions about the nature of the relationship between acoustic parameters and perceived prosody.

Our hope is to demonstrate that since all of our voice building processes are fully automatic, and we do not rely on any manual pitch-marking, segmentation or labeling processes, Lessac techniques can be used to build multiple near natural human sounding synthetic voices quickly.

Participating in the Blizzard Challenge has proven very helpful for us in guiding further improvements of our technologies.

6. References

[1] Lessac, A., The Use and Training of the Human Voice: A Bio-Dynamic Approach to Vocal Life, McGraw-Hill, 1996.

[2] Nitisaroj, R. and Marple, G. A., "Use of Lessemes in text-to-speech synthesis", in M. Munro, S. Turner, A. Munro, and K. Campbell [Eds], Collective Writings on the Lessac Voice and Body Work: A Festschrift, Llumina Press, 2010.

[3] Jordan, M. I. and Jacobs, R. A., "Hierarchical Mixtures of Experts and the EM Algorithm", Neural Computation, 6:181-214, 1994.

[4] Ma, J., Xu, L. and Jordan, M. I., "Asymptotic Convergence Rate of the EM Algorithm for Gaussian Mixtures", Neural Computation, 12:2881-2900, 2000.

[5] Kominek, J., Bennett, C., Langner, B. and Toth, A., "The Blizzard Callenge 2005 CMU Entry: A Method for Improving Speech Synthesis Systems", Proceedings of Interspeech 2005, 85-88.