Analysis of a low-dimensional bottleneck neural network representation of speech for modelling speech dynamics

Linxue Bai, Peter Jančovič, Martin Russell, Philip Weber

School of Electronic Electrical and Systems Engineering,
The University of Birmingham, Birmingham B15 2TT, UK
{lxb190, p.jancovic, m.j.russell}@bham.ac.uk,phil.weber@bcs.org.uk

Abstract

This paper presents an analysis of a low-dimensional representation of speech for modelling speech dynamics, extracted using bottleneck neural networks. The input to the neural network is a set of spectral feature vectors. We explore the effect of various designs and training of the network, such as varying the size of context in the input layer, size of the bottleneck and other hidden layers, and using input reconstruction or phone posteriors as targets. Experiments are performed on TIMIT. The bottleneck features are employed in a conventional HMM-based phoneme recognition system, with recognition accuracy of 70.6% on the core test achieved using only 9-dimensional features. We also analyse how the bottleneck features fit the assumptions of dynamic models of speech. Specifically, we employ the continuous-state hidden Markov model (CS-HMM), which considers speech as a sequence of dwell and transition regions. We demonstrate that the bottleneck features preserve well the trajectory continuity over time and can provide a suitable representation for CS-HMM.

Index Terms: modelling speech dynamics, continuous-state hidden Markov model, neural networks, bottleneck features, low-dimensional features, formants, TIMIT

1. Introduction

A conventional hidden Markov model (HMM) models the speech signal as a sequence of piece-wise constant segments. The information about dynamics of speech is typically incorporated into the feature representation by concatenating features describing the current signal frame ('static') with their time derivatives ('delta'). Over the years, there has been considerable interest in alternative models which aim to model speech dynamics more accurately (see [1] for a review). In these models, which we refer to as segmental, the states are associated with sequences of acoustic vectors, or segments, rather than with individual acoustic feature vectors as in conventional HMMs. A recent addition to these models is the continuous-state HMM [8, 9], which can be applied to general segmental models of speech such as the Holmes-Mattingly-Shearme dwell-transition model [10]. We use this model in this paper.

Mel-frequency cepstral coefficients (MFCCs) are currently the mainstream acoustic feature representation. Although MFCCs have been shown to perform well for speech recognition, they are less suitable as feature representation for segmental models of speech dynamics. This is due to the fact that in representations of speech derived through a linear transformation of short-term spectra the articulator dynamics of speech are manifested indirectly, often as movement between, rather than within, frequency bands. Representation of speech in terms of formant parameters (frequencies and amplitudes) or articulatory features directly describes the process of speech production and preserves speech dynamics. However, formant features are notoriously difficult to estimate reliably. Moreover, they are not well defined for some speech sounds. Therefore, there is a need for compact representations of speech that can be reliably estimated for all speech sounds.

Over the last few years, there has been an intensive research interest on employing (deep) neural networks (NNs) for speech recognition. A variety of ways of using the NNs have been investigated. One of the main approaches is to use NNs as a non-linear feature extractor. Features derived from the output layer, e.g. [11], or various intermediate hidden layers of the NNs, e.g. [12], can be used. Comparisons between different deep NN hidden and output layer features as well as their concatenations were studied in [15]. Bottleneck features (BN) are a special form of NN features which are extracted from NNs with a compression layer. The bottleneck structure provides a way to reduce dimensionality. In recent research, most BN-NNs have tens to hundreds of neurons at the bottle-neck layer, such as [20, 21].

In this paper, we analyse the suitability of low-dimensional features extracted from the bottleneck of neural networks, for modelling speech dynamics. We use neural networks having five layers. Logarithm filter-bank energies with context are used as input to the network. The output of the middle bottleneck layer is used as feature representation. We first assess various ways of designing and training the network. This includes varying the size of bottleneck or intermediate hidden layers, the use of the input spectral features (reconstruction) or phone posteriors as targets for training the network. We then assess the suitability of the obtained bottleneck features in a CS-HMM system. Experimental evaluations are performed on the TIMIT speech corpus [13]. We demonstrate that the low-dimensional bottleneck features thus obtained give on average 33.7% reduction in phone errors compared with formant-based features of the same dimensionality in a conventional HMM-based ASR system. We observe that the bottleneck features preserve better the trajectory continuity and fit better the CS-HMM modelling than formants. The bottleneck features provide a compact representation in terms of the number of model parameters and they seem to be in overall well suited to be employed for segmental models of speech dynamics.

2. Modelling speech dynamics

Considerable research effort has been devoted to developing models of speech dynamics which more faithfully reflect the properties of speech structure than conventional HMMs. Such

dynamic models of speech aim in various ways to reduce the assumptions that speech is a piece-wise stationary process and that the observations are temporally independent, as well as improve duration modelling. A comprehensive survey of many different types of statistical models of speech dynamics is given in [1]. This includes segmental HMMs [3], trajectory models [2, 4, 5], intermediate state models [6], Gaussian process dynamical models [7] and more recently continuous-state HMMs (CS-HMMs) [8, 9]. In this paper, we employ the continuous-state HMM (CS-HMM), briefly outlined next.

We assume speech to fit the Holmes-Mattingly-Shearme model, in which dwells represent phoneme targets and transitions the smooth migration from one dwell to the next. d-dimensional input features are assumed to be noisily distributed around underlying dwell 'realisations' for a phoneme instance, or around transitions. Dwell realisations are also assumed to be noisily distributed around reference targets for each phoneme.

A sequential branching algorithm is used to recover a sequence of alternating dwells and transitions, the times of changes between them, and the sequence of phonemes which could have generated them. A set of hypotheses is maintained for possible trajectories. Each maintains a 'state' consisting of Baum-Welch alpha values $\alpha_t(\boldsymbol{x})$ in the form of a scaled Gaussian distribution representing its belief of the current underlying target values which generated the observed data, together with the 'slope' of the current segment. These are updated following each observation. Discrete components of the state store the current phoneme identity, time since previous phoneme, and phoneme history (for a language model).

Following each observation all hypotheses are split, to model the alternatives of continuing the current dwell or transition, or changing from dwell to transition or vice versa. For every new dwell, a hypothesis is created for every phoneme in the inventory, while for a new transition a d-dimensional vector of slope values is appended to $\boldsymbol{x}$, and marginalised out at the end of the transition. Low probability hypotheses are 'pruned' to maintain computational efficiency. For full details of the algorithm and update calculations we refer the reader to [9].

In this work we use a CS-HMM to recover the dwell-transition trajectory that best fits the features, ignoring phoneme targets. To do this we supply the model with an inventory with a flat prior, consisting of a single phoneme target with very high variance. We return the top hypothesis with the same number of dwells as there are TIMIT labelled phonemes in the utterance. Note that a distinguishing feature of the dwell-transition CS-HMM used in this work is that continuity is preserved across the segment boundaries.

3. Representing speech using bottleneck neural network features

One of the problems with using models of speech dynamics, as mentioned in Section 2, is the requirement of adequate feature representations of the acoustic signal. The extracted features are expected to have a smooth trajectory over time. This is not valid for conventional mel-frequency cepstral coefficients. As such a different representation of speech is needed. The most natural is to use a formant representation of speech. However, formants are difficult to estimate reliably and they are not well defined for some speech sounds. Over the last few years, there has been a huge popularity in (deep) neural networks (NN). As a NN performs a non-linear mapping, they seem a natural way to employ for our problem of representing speech.

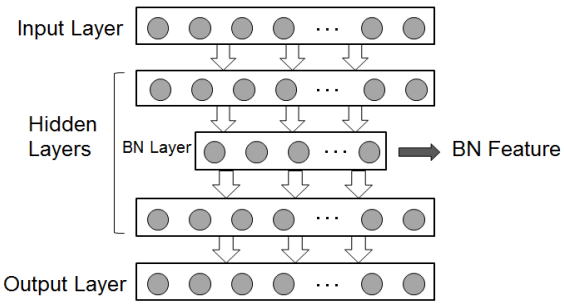


Figure 1: *Architecture of the multi-layer bottleneck neural network employed for speech representation.*

The architecture of the bottleneck neural network that we used is depicted in Figure 1. The input to network is a vector containing the logarithm filter-bank energies (logFBEs) of the current signal frame and several preceeding and following frames. We explored two approaches to training the network, one aimed at the reconstruction of the input spectrum and the other at discrimination between phonemes. In the case of reconstruction, the target is a vector of logFBEs of the current signal frame. In the case of phoneme discrimination, the targets were the posterior probabilities of the 49 phones. These were obtained based on the labels and time-stamp information supplied with TIMIT. A mapping of 61 to 49 phones was performed as described in [16]. The Softmax function was used at the last layer.

4. Experimental setup

4.1. Speech corpus

Experiments were performed on the TIMIT speech corpus [13]. We used the training set, containing 462 speakers, without the SA recordings to train the models. The reported results are based on the core test set, containing speech from 24 speakers.

4.2. Feature extraction based on bottleneck neural network

The speech, sampled at 16 kHz, was analysed using a 25-ms Hamming window with a 10-ms frame rate. We used Mel-frequency filter-bank analysis, implemented based on the Fourier transform. The number of filter-bank channels was set to 26, covering the range from 0 to 8 kHz. The logarithm filter-bank energies (logFBEs) were normalised to have zero mean and unit variance based on the entire training set.

We used 90% of the training set (random 90% of utterances for each gender in each dialect) as the neural network training set, and the remaining 10% as validation set. The networks were trained with stochastic gradient decent using the Theano toolkit [18, 19]. The maximum epoch number was set to 3000. The training stopped when the error on the validation set started to rise or when the epoch reached the maximum.

4.3. HMM-based phoneme recognition system

Speech recognition experiments were performed using a standard GMM-HMM system built using HTK [14]. HMMs were built for the 49 phone set, each model consisting of 3 states. The number of GMM components per state was set to increase from 1 in powers of 2 up to 512. The number of components for silence was twice of that for phonemes. A bigram language

model was used. For evaluating recognition performance, the 49 phone set was reduced to 40 according to [17] . The reported results are on the core test set using the number of GMM components corresponding to the best accuracy achieved on the validation test set.

5. Experimental results

5.1. Analysis of the bottleneck representation of speech

This section presents results demonstrating the effect of different designs of the neural network. The bottleneck features are employed in a conventional HMM-based ASR system and the phone recognition accuracy is reported.

Our initial experiments explored the use of the network for spectrum reconstruction or phoneme-probability estimation. The best result achieved using the reconstruction network was 56.8% phone accuracy, obtained with 16 bottleneck features and 128 units in the other hidden layers, while the phoneme-posterior network with the same number of neurons in the hidden layers achieved phone accuracy of 69.7%. The inclusion of context in the input layer had negligible effect on the performance in the case of the spectrum reconstruction network. Based on this, all results reported in the remainder of the paper are using the network with the phoneme-probability as targets.

Next, we explored the effect of varying the number of neurons in hidden layers of the network with the phoneme-posteriors as targets. We used 3 hidden layers, denoted as H-B-H, where 'B' stands for the bottleneck layer. The number of neurons in both 'H' hidden layers was kept the same and set to 32, 128, 512, 1024, and 2048. The number of neurons in the bottleneck layer was set to 4, 9, 16, and 32. Our experiments showed that best performance was obtained using the context of 5 frames around the current frame in the input layer and as such only those results are reported here, i.e., the input layer is of size 286. The output layer is of size 49. The results achieved by monophone HMMs are depicted in Figure 2. It can be seen that a considerable performance improvement is obtained when the bottleneck layer increases from 4 to 9 neurons. Increasing the bottleneck further beyond 9 neurons gives only minor improvements. Increasing the size of the other two hidden 'H' layers from 32 to 512 also gives a great improvement in performance, but only minor improvements are seen when the size is above 512.

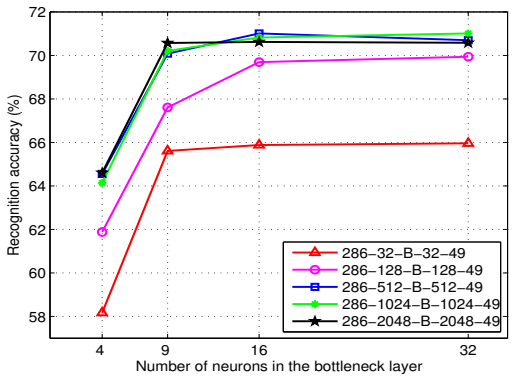


Figure 2: *Phone recognition accuracy using bottleneck features as a function of the number of neurons in the bottleneck and other hidden layers when using phoneme-posterior network.*

Figure 3 shows the performance with bottleneck features obtained from the phoneme-posterior network when using monophone and triphone modelling and adding delta features. The network configuration 286-512-B-512-49 was used, with varying the bottleneck layer size. It can be seen that by appending Δ and $\Delta\Delta$ to bottleneck features gives improvement between 2% to 4%, depending on the size of bottleneck. Note that the results obtained using 9 dimensional bottleneck features are better than using 4 dimensional bottleneck features appended by Δ and $\Delta\Delta$, i.e., 12 dimensional features. This suggests that the bottleneck features are containing both the spectral and temporal properties of speech. Interestingly, the use of triphone models gives lower performance than monophone models, especially for very low-dimensional bottleneck features. This suggests that the contextual information may have been compressed due to the very low-feature dimensionality.

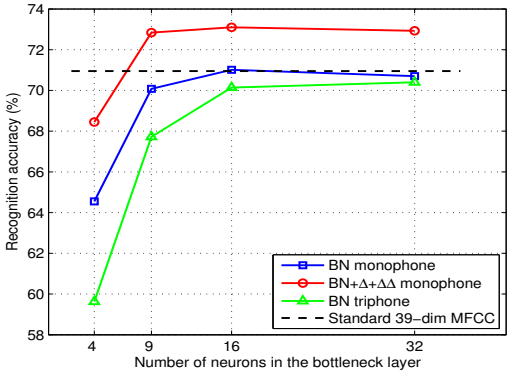
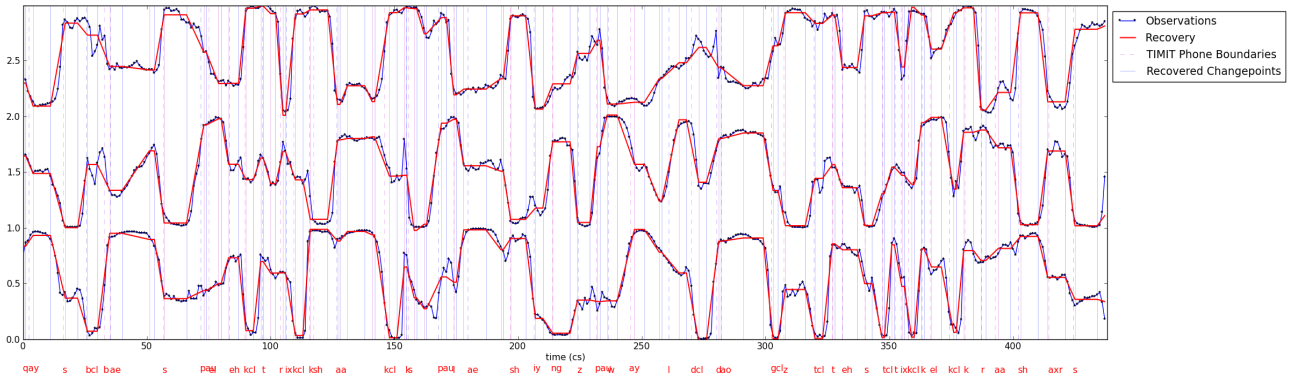


Figure 3: *Phone recognition accuracy using bottleneck features extracted from phoneme-posterior network 286-512-B-512-49 when varying the size of the bottleneck layer.*

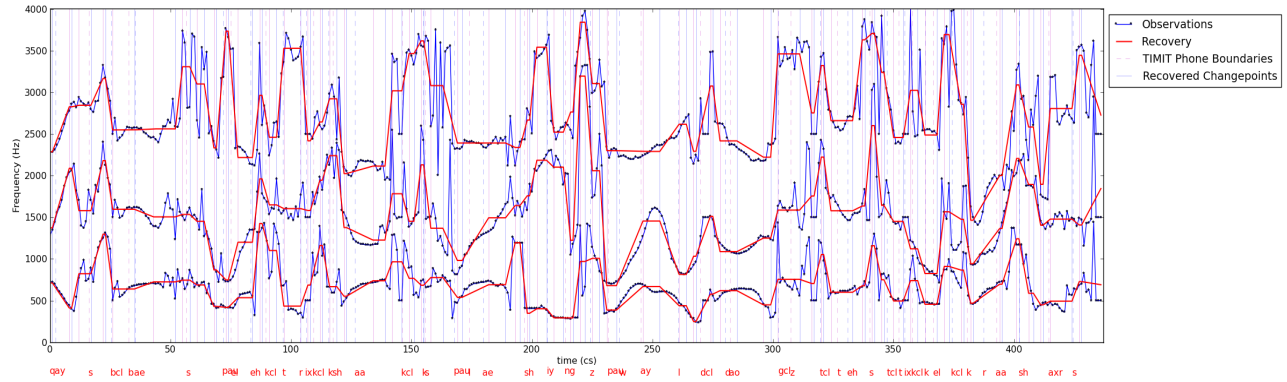
5.2. Analysis of the bottleneck features for modelling speech dynamics

This section analyses the use of bottleneck features for modelling speech dynamics, with comparisons made to formant-based representation.

First, we compare the performance of the bottleneck and formant features when using a conventional HMM-based ASR system. Results are presented in Table 1. Experiments were performed with formants estimated using the Wavesurfer [23] and Praat [22] tool. The former achieved better results and as such only these are reported here. It can be seen that the use of 3 bottleneck features considerably outperforms the use of 3 formant frequencies. It seems that the bottleneck features may be able to encode information about both frequency and amplitude. Thus, we also performed experiments with formant-based features containing the formant frequencies, amplitudes and bandwidths, resulting in a 27 dimensional feature vector (with delta and delta-delta) and compared these with the same dimensionality bottleneck features (with delta and delta-delta). It can be seen that bottleneck features considerably outperformed formant-based features. The use of the bottleneck-based feature representation results, on average, in a 33.7% reduction in phone errors compared with formant-based features with the same dimension. The confusion matrices recognition results showed that the bottleneck features achieved nearly uniform improvement over formant-based features across all phonemes.



(a)



(b)

Figure 4: *An example of the dwell-transition trajectories recovered by the CS-HMM when using bottleneck features (a) and estimated formant frequencies (b). Blue line with dots show the observations (feature values) and solid red line the estimated dwell-transition trajectory. TIMIT phoneme boundaries are indicated by thin vertical lines, recovered dwell starts (magenta) and ends (blue) by vertical dashed lines.*

Table 1: *Recognition performance of an HMM-based ASR system when using formant and bottleneck feature representation.*

Feature representation	Dim.	Recognition	
		Corr (%)	Acc (%)
Baseline: MFCC + Δ + ΔΔ	39	76.23	70.95
Formants			
3 freq	3	49.30	40.71
3 freq + Δ + ΔΔ	9	56.32	51.12
3 freq & amp & bw	9	55.96	52.04
3 freq & amp & bw+ Δ + ΔΔ	27	65.06	60.43
Bottleneck features			
3 BN features	3	65.02	60.94
3 BN features + Δ + ΔΔ	9	70.87	65.73
9 BN features	9	74.37	70.57
9 BN features + Δ + ΔΔ	27	76.77	73.07

Now, we analyse the performance when using the continuous-state HMM (CS-HMM) for modelling speech dynamics. Figures 4 shows an example of trajectories recovered by the CS-HMM when using 3 bottleneck features and formant frequencies, respectively. The formants range from 0 to 4000 Hz, while the bottleneck features are in the range [0, 1] but plotted shifted on the vertical axis for clarity. These plots suggest that bottleneck features fit the model considerably better than the formant features. The formant trajectories seem smooth in voiced regions but vary widely in unvoiced regions and this seems to affect their fit to the model.

6. Summary and future work

Segmental models of speech hold promise for speech recognition due to their ability to parsimoniously model speech dynamics. However they have been hampered by lack of a good representation. Formants model voiced sounds well, but are inappropriate for unvoiced speech, while articulatory parameters are difficult to obtain. We presented results of ASR experiments using low-dimensional bottleneck features extracted from Neural Networks, and an initial analysis of their temporal dynamics. Our results showed that when the networks were trained to predict phoneme posteriors, bottleneck features significantly outperformed formant features of similar dimensionality. We plan to extend this work to understand the characteristics of speech being captured by bottleneck features and how to influence the features through the training of the Neural Networks. We also plan to integrate these features into our CS-HMM recognisers [8, 9].

7. References

[1] L. Deng, *Dynamic Speech Models, Theory, Algorithms, and Applications*. Morgan & Claypool, 2006.

[2] L. Deng and J. Ma, "Spontaneous speech recognition using a statistical coarticulatory model for the vocal-tract-resonance dynamics," *J. Acoust. Soc. Am.*, vol. 108, no. 6, pp. 3036–3048, 2000.

[3] M. Ostendorf and V. Digalakis and O. Kimball, "From HMM's to segment models: a unified view of stochastic modeling for speech recognition," *IEEE Trans. Speech Audio Proc.*, vol. 4, no. 5, pp. 360–378, 1996.

[4] M. J. F. Gales and S. J. Young, "Segmental hidden Markov models," in *Proc. Eurospeech*. ISCA, 1993.

[5] H. B. Richards and J. S. Bridle, "The HDM: a segmental hidden dynamic model of coarticulation," in *Proc. ICASSP*, Piscataway, USA, 1999, vol. 1, pp. 357–60.

[6] G. E. Henter and W. B. Kleijn, "Intermediate-State HMMs to capture continuously-changing signal features," in *Proc. Interspeech*, Florence, Italy, 2011, vol.12, pp. 1817 – 20.

[7] G. E. Henter, M. R. Frean and W. B. Kleijn, "Gaussian process dynamical models for nonparametric speech representation and synthesis," in *Proc. ICASSP*, Kyoto, Japan, 2012, pp. 4505 – 4508.

[8] P. Weber, S. M. Houghton, C. J. Champion, M. J. Russell, and P. Jančovič, "Trajectory analysis of speech using continuous state hidden Markov models," in *Proc. ICASSP*, pp. 3042–3046, Florence, Italy, 2014.

[9] C. J. Champion and S. M. Houghton, "Application of Continuous State Hidden Markov Models to a classical problem in speech recognition," *Computer Speech and Language*, submitted, 2014.

[10] J. N. Holmes, I. G Mattingly, and J. N. Shearme, "Speech synthesis by rule," *Language and speech*, 7(3), pp. 127–143, 1964.

[11] H. Hermansky, D. P. W. Ellis, and S. Sharma, "Tandem connectionist feature extraction for conventional HMM systems," in *Proc. ICASSP*, pp. 1635–1638, Istanbul, Turkey, 2000.

[12] F. Grézl, M. Karafiát, and M. Janda, "Study of probabilistic and bottle-neck features in multilingual environment," in *Proc. ASRU*, pp. 359–364, Waikoloa, HI, USA, 2011.

[13] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, D. S. Pallett, and N. L. Dahlgren, "Darpa TIMIT acoustic-phonetic continuous speech corpus CD-ROM," NIST, Tech. Rep., 1990.

[14] S. J. Young, J. Odell, D. Ollason, and V. Valtchev, P. Woodland, "The HTK Book," Entropic Camb. Res. Lab, Cambridge, UK, 1997.

[15] L. Deng, J. Chen, "Sequence classification using the high-level features extracted from deep neural networks," in *Proc. ICASSP*, pp. 6844–6848, Florence, Italy, 2014.

[16] W. J. Holmes, "Modelling Segmental Variability for Automatic Speech Recognition," *PhD thesis*, University of London, 1997.

[17] K. F. Lee, H. W. Hon, "Speaker-independent phone recognition using hidden Markov models," *IEEE Trans. Acoustics, Speech, and Signal Proc.*, vol. 37, pp. 1641–1648, 1989.

[18] Bastien, Frédéric and Lamblin, Pascal and Pascanu, Razvan and Bergstra, James and Goodfellow, Ian J. and Bergeron, Arnaud and Bouchard, Nicolas and Bengio, Yoshua, "Theano: new features and speed improvements," in *Proc. Deep Learning and Unsupervised Feature Learning Workshop*, 2012.

[19] Bergstra, James and Breuleux, Olivier and Bastien, Frédéric and Lamblin, Pascal and Pascanu, Razvan and Desjardins, Guillaume and Turian, Joseph and Warde-Farley, David and Bengio, Yoshua, "Theano: a CPU and GPU Math Expression Compiler," in *Proc. of the Python for Scientific Computing Conference (SciPy)*, Austin, TX, June 2010.

[20] Grezl, Frantisek and Karafiát, Martin and Kontár, Stanislav and Cernockỳ, Jan "Probabilistic and Bottle-Neck Features for LVCSR of Meetings," in *Proc. ICASSP*, pp. 757–760, 2007.

[21] Liu, Diyuan and Wei, Si and Guo, Wu and Bao, Yebo and Xiong, Shifu and Dai, Lirong "Lattice based optimization of bottleneck feature extractor with linear transformation," in *Proc. ICASSP*, Florence, Italy, 2014.

[22] P. Boersma, D. Weenink, "Praat: doing phonetics by computer," *Computer program, retrieved 15 September 2013 from http://www.praat.org/*, 2013.

[23] K. Sjolander and J. Beskow, "Wavesurfer – an open source speech tool," in *Proc. Interspeech*, pp.464–467, 2000.