On the impact of phoneme alignment in DNN-based speech synthesis

Mei Li[1], *Zhizheng Wu*[2], *Lei Xie*[1]

[1]Shaanxi Provincial Key Laboratory of Speech and Image Information Processing,
School of Computer Science, Northwestern Polytechnical University, Xi'an, China
[2]The Centre for Speech Technology Research (CSTR), The University of Edinburgh, U.K.
{meili,lxie}@nwpu-aslp.org, zhizheng.wu@ed.ac.uk

Abstract

Recently, deep neural networks (DNNs) have significantly improved the performance of acoustic modeling in statistical parametric speech synthesis (SPSS). However, in current implementations, when training a DNN-based speech synthesis system, phonetic transcripts are required to be aligned with the corresponding speech frames to obtain the phonetic segmentation, called phoneme alignment. Such an alignment is usually obtained by forced alignment based on hidden Markov models (HMMs) since manual alignment is labor-intensive and time-consuming. In this work, we study the impact of phoneme alignment on the DNN-based speech synthesis system. Specifically, we compare the performances of different DNN-based speech synthesis systems, which use manual alignment and HMM-based forced alignment from three types of labels: HMM mono-phone, tri-phone and full-context. Objective and subjective evaluations are conducted in term of the naturalness of synthesized speech to compare the performances of different alignments.

Index Terms: Speech synthesis, acoustic modeling, deep neural networks, phoneme alignment

1. Introduction

There have been many efforts dedicated to the synthesis of natural-sounding speech. As a popular category of methods, statistical parametric speech synthesis (SPSS) has been advanced significantly recently. SPSS offers greater flexibility and controllability than the unit-selection or waveform-concatenation method [1]. These abilities mainly come from the flexibilities of the underlying small-footprint acoustic models, which map linguistic features to acoustic speech parameters for waveform generation.

As an elegant sequential SPSS model, hidden Markov models (HMMs) have dominated the acoustic modeling in the past decade [2]. In a typical HMM system, a multi-space probability distribution HMM (MSD-HMM) [3] is used to model the spectrum, pitch and duration simultaneously at the state-level [4]. A decision tree based context clustering strategy is adopted to handle unseen linguistic features when mapping from the rich-context linguistic features to the acoustic features. As a result, HMM parameters are shared across the clustered groups of linguistic contexts.

Despite years of efforts, the naturalness of the HMM-synthesized speech is still unable to compete with good unit selection synthesizers. A major factor that degrades the naturalness is the accuracy of the acoustic models [1]. Recently, neural networks have re-emerged as a powerful tool for acoustic modeling in SPSS [5, 6, 7, 8, 9, 10, 11, 12], following the success of deep learning in speech recognition [13] and many other machine learning tasks. Through a deep neural network (DNN), a frame-level regression model is directly learned from linguistic labels to acoustic features without a decision tree. DNN-based acoustic models provide an efficient and distributed representation of complex dependencies between linguistic and acoustic features [5]. A number of studies have demonstrated that DNNs are able to achieve significantly better performances than decision-tree based HMMs [5, 6, 12]. Some network variants, eg., DNNs with multi-task learning and stacked bottleneck features [7], deep mixture density networks (MDN) [8], long short-term memory (LSTM) recurrent network [9] and its simplified versions [10], have shown their potential to produce more natural-sounding synthesized speech. A very recent study [12] has confirmed that two critical factors, replacing decision trees with DNNs and moving from state-level to frame-level predictions, contribute much to the improvement of the naturalness of a DNN system.

Different from HMM-based synthesis, which could start without phoneme alignment, training a DNN-based acoustic modeling needs phoneme alignment or frame-level phonetic segmentation information (at least in current implementations). As Watts *et al.* pointed out [12], frame-level prediction secures the performance gain of a DNN-based acoustic model. However, how does the phoneme alignment affect the performance of DNN-based synthesis is still unknown. Even though the phoneme alignment can be obtained manually, it is not practical for large corpora. On the other hand, HMM-based model, which does not require phoneme alignment to train, may provide an alternative “cheap” way. To this end, we look into the impact of phoneme alignment on the training of DNN-based acoustic models in this paper. We use the manual alignment to benchmark the performance and choose three forced alignment methods for comparison: HMM mono-phone, tri-phone and full-context models.

2. Related works

In some TTS systems, especially for commercial synthesizers, manual alignment has been employed since it is considered as most reliable and precise way to get the frame-level phonetic segmentation information [14]. However, manual alignment is time-consuming and labor-intensive [15, 16]. It is not practical for large corpora. Instead, HMM-based forcing alignment has been pointed out as the most practical method for automatic phonetic segmentation.

Several studies have been carried out in investigating the performance of different forcing alignment methods [17, 18, 19, 20]. According to their results, a simple forcing alignment model (e.g., HMM mono-phone model) is able to achieve similar or even better performance than that of a more complex model (e.g., HMM tri-phone model or HMM full-context model). Some papers have focused on the improvmment of alignment accuracy [21, 22]. On the other hand, the impact of alignment on the unit-selection synthesis has also been studied [23, 14]. In [23], a regression tree based on phonetic boundaries was used to conduct boundary specific correction to refine the HMM-based segmentation. Alignment accuracy comparable to manual alignment was thus obtained. However, this study has shown that the improvement on alignment does not carry obvious benefits to the synthesis performances. In [14], Chu *et al.* conducted manual check on the forced alignment results and experiments showed improvements on the naturalness of synthesized speech.

Nevertheless, improving the phonetic segmentation performance is out of the scope of this paper. In this paper, we are interested in investigating the impact of different alignment methods (manual and forced alignment) on the performance of DNN-based speech synthesis.

3. Phonetic Alignment

3.1. Manual Alignment

Usually, it needs several language experts continuously working several days to get the manual alignment for a sizable TTS corpus [19]. Moreover, in some cases, the alignment results are different across the experts. Thus, in order to ensure that the manual alignment is reliable, the labeling difference between experts should be smaller than a certain threshold. In this study, the manual alignment of our corpus is obtained in this way and three language experts are paid to do the phonetic segmentation.

3.2. Forcing Alignment

Forcing alignment methods have been widely adopted. It allows the alignment to be both consistent and reproducible at a very low cost. One of the most frequently used approaches for forcing alignment is based on HMM using the HTK toolkit [24]. A set of phonetic HMMs are first trained using a training set. Then with the phoneme transcriptions and the corresponding wave files provided, a Viterbi search algorithm is used for the alignment and transitions between different HMM models are then considered as phoneme boundaries.

In this study, three versions of forcing alignment methods are investigated from the following HMM models:

- Mono_Nmix: mono-phone HMMs with N Gaussian mixtures per state;
- Tri_Nmix: tri-phone HMMs with N Gaussian mixtures per state;
- Full_Nmix, full-context HMMs [1] with N Gaussian mixtures per state.

We investigate the impact of context dependencies and the number of Gaussian mixtures on the performance of the HMM-based forcing alignment. We first train a set of mono-phone HMMs with five-state, left-to-right model topology, where each state is modeled by a single Gaussian, diagonal covariance output distribution. For tri-phone and full-context models, the decision tree based context clustering is performed using their corresponding question set. An unseen model is trained in case that the development or testing set contains new phonetic models that the training corpus does not contain. This is achieved by assigning class average values to non-existing ones. The number of Gaussian mixtures in the three sets of HMMs is set from 1 to 32 to test their influences.

4. Experimental Setups

4.1. Corpus

A phonetically-balanced corpus with 5,470 Chinese sentences (3-5 seconds per sentence) spoken by a native female speaker in neutral style was used in our experiments. The corpus had phonetic transcripts with manually labeled phoneme boundaries. We randomly selected 5,000 sentences for model training, 270 for development and 200 for testing in the experiments. Speech waveforms are sampled at 16kHz, windowed by a 25ms window shifted

[1] The full label [25] includes quin-phone, the position of phone, syllable and word in phrase and sentence, the length of word and phrase, stress of syllable, TOBI and POS of word.

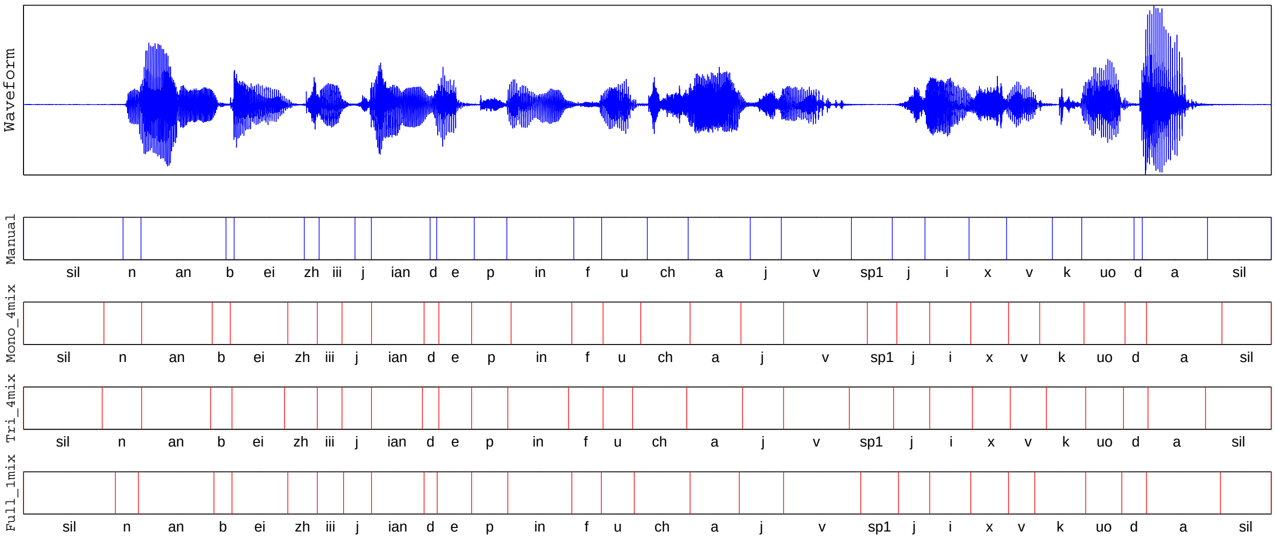

Figure 1: The phoneme alignment results for a sentence in the training set.

every 5ms in the acoustic feature extraction and modeling.

4.2. HMM-based forcing Alignment

In the HMM-based forcing alignment, the feature vector for model training is composed of 18 dimensional Mel-frequency cepstral coefficients (MFCCs), normalized log energy, as well as their delta and delta-delta components (57-dimension in total). Please note that we only use 5,000 training sentences to train the HMM models for forcing alignment and then use the trained models to align all the 5,470 sentences. Phoneme alignment was carried out according to the methods described in Section 3.2.

4.3. DNN Model Training

We train speaker-dependent DNN acoustic models using different phoneme alignment results to investigate their influences on a DNN synthesizer. The input feature vector of the network contains 657 dimensions, where 596 dimensions are binary features for categorical linguistic contexts, 58 dimensions are numerical linguistic contexts and the rest 3 dimensions are numerical features that inlcude the frame positions to the start and the end of the current phoneme and the frame numbers. The output acoustic parameters include 41-dimensional LSPs and linearly interpolated F_0 in log-scale with their delta, delta-delta features, plus a voicing/unvoicing (V/UV) flag, totally 127 dimensions. In the training, 80% of the silence frames are removed from the training data to avoid DNN over-learning the silence label. The input linguistic features are normalized to a fixed range [0.01 0.99] and the output acoustic features are normalized by mean-variance normalization (MVN). The acoustic model is a feed-forward DNN with 6 hidden layers of 1024 nodes in each layer. The *tanh* and linear activation functions are used for the hidden layers and the output layer, respectively. The hyper-parameters, such as learning rate and momentum, are tuned on the development set. The DNN training procedure is implemented in Python using the Theano toolkit [26]. Finally, the waveforms are synthesized by an LPC synthesizer using the predicted speech parameters. We investigate the impact from different phoneme alignments by changing the input vector of the DNN, i.e., the last 3 dimensions of the network input vector that reflect different phoneme alignment results.

5. Experimental Results

5.1. Phoneme Alignment Accuracy

Figure 1 shows the phoneme alignment results from different methods for a sentence in the corpus. We can clearly see the difference among various alignment methods. In the experiments, we first compare the phoneme alignment accuracy of different alignment methods. If the distance from a forced alignment phoneme boundary to it's manual reference is smaller than a certain tolerance threshold, it is counted as a correct one. The accuracy scores (from 10 to 40ms threshold) on the training set are reported in Table 1.

From Table 1, we can see that the best alignment results are achieved from different numbers of Gaussian mixtures for different tolerance windows. We notice that for smaller tolerance windows (10 and 20ms), HMMs with 4 Gaussian mixtures achieve the highest accuracy for mono-phone and tri-phone while HMMs with 2 Gaussian mixtures achieve the highest accuracy for full-

Table 1: *Alignment accuracies on the training set with different tolerance values (10, 20, 30, 40 ms) using mono-phone, tri-phone and full-context HMMs with different number of Gaussian mixtures per state.*

Model Set	10	20	30	40
Mono_1mix	30.02	57.53	**76.48**	86.70
Mono_2mix	32.96	59.13	76.34	86.94
Mono_4mix	**33.11**	**59.26**	76.43	**87.24**
Mono_8mix	32.81	58.50	75.75	86.92
Mono_16mix	32.37	57.21	74.79	86.16
Mono_32mix	31.59	55.87	73.89	85.52
Tri_1mix	29.97	56.14	**73.74**	**84.31**
Tri_2mix	30.46	56.63	73.15	83.85
Tri_4mix	**30.53**	**56.73**	72.71	83.60
Tri_8mix	30.44	56.67	72.40	83.32
Tri_16mix	30.36	56.56	72.23	83.21
Tri_32mix	30.19	56.39	72.07	83.17
Full_1mix	32.31	59.14	**77.57**	**87.53**
Full_2mix	**32.52**	**59.31**	75.60	86.51
Full_4mix	31.90	58.67	75.06	85.79
Full_8mix	29.55	56.81	72.81	83.48
Full_16mix	26.50	53.79	70.55	81.31
Full_32mix	23.96	49.96	63.70	78.38

context. In general, tri-phone models are not as good as mono-phone. This result is consistent with the conclusion drawn in [17]: the segmentation produced by context-dependent HMMs tend to be less precise than the ones produced by context-independent HMMs. A theoretical explanation for this behavior can be found in [27]. As another set of context-dependent models, full-context HMMs, achieve segmentation results as good as mono-phones. The accuracy may come from the prosodic information provided in the full-context labels, as explained in [19].

5.2. Speech Synthesis Objective Evaluation

We choose Mono_4mix, Tri_4mix and Full_1mix for the evaluation of DNN-based speech synthesis. We first analyse the impact of different types of phoneme alignments on the DNN training using the testing set with manual alignment (i.e., the mismatched case). Results are shown in Table 2. The results show that there is a clear gap between the forced alignment labels (Mono_4mix, Tri_4mix and Full_1mix) and the manual alignment labels (Manual). This is because of the phoneme alignment mismatch between the training and synthesis stages. The mono-phone with 4 Gaussian mixtures obtains lowest LSD, F0 and UV prediction errors among the three systems using forced alignment to train the DNN model.

We then analyse the 4 different kind of DNN synthesizers using their corresponding phoneme alignment labels respectively in the testing set (i.e., the matched case). The results are presented in Table 3. The results show that the gaps between the forced alignment labels (Mono_4mix, Tri_4mix and Full_1mix) and manual alignment labels (Manual) are not salient any more. But we should notice that the acoustic features produced by the different phoneme alignments may have different frame numbers, thus the numbers in Table 3 are not directly comparable.

Table 2: *Log-spectral distance (LSD), root mean squared errors (RMSEs) of F_0, voiced/unvoiced error rates (V/UV) on the testing set with manual alignment for 4 different forced alignment trained DNNs (the mismatched case)*

Model Sets	LSD (dB)	F_0 RMSE (Hz)	V/UV (%)
Manual	2.2666	26.670	8.085
Mono_4mix	**2.6076**	**28.142**	**12.511**
Tri_4mix	2.6368	28.345	13.289
Full_1mix	2.6357	28.484	12.864

Table 3: *Log-spectral distance (LSD), root mean squared errors (RMSEs) of F_0, voiced/unvoiced error rates (V/UV) on the testing set with their corresponding phoneme alignment for 4 different forced alignment trained DNNs (the matched case).*

Model Sets	LSD (dB)	F_0 RMSE (Hz)	V/UV (%)
Manual	2.2666	26.670	8.085
Mono_4mix	2.2725	27.228	8.599
Tri_4mix	**2.2716**	**26.667**	8.355
Full_1mix	2.2801	27.238	**8.304**

5.3. Speech Synthesis Subjective Evaluation

In order to see if the human perception is consistent with the objective evaluation, we conduct two subjective experiments to assess the naturalness of the synthesized speech using a MUSHRA (MUltiple Stimuli with Hidden Reference and Anchor) test [28]. In each experiment, subjects are asked to rate, using a scale from 0 (completely bad) to 100 (completely natural), on 20 sets of stimuli randomly chosen from the testing set. Each set of stimuli represents the same sentence, but synthesized using the four different DNN systems and a matching copy-synthesized one as hidden reference (i.e., the anchor). Stimuli are ordered randomly and presented without labels. Subjects are instructed to rate the hidden reference as completely natural, fixing the high end of the scale. No explicit lower anchor is used, since the synthetic speech stimuli themselves are sufficiently different from the copy-synthesized speech to act as implicit anchors. A group of 30 native Chinese listeners participated in the test. In total, 600 sets of parallel rating are obtained for each experiment.

The distributions of subjective ratings on the testing set using manual alignment for 4 different phoneme alignment trained DNNs are shown in Figure 2, while the distributions of subjective ratings on the testing set using their corresponding matched alignment for 4 different phoneme alignment trained DNNs are shown in Figure 3. Box edges are at 25% and 75% quantiles. Red line is the median and green dashed line is the mean. From Figure 2, it is clear that the copy-synthesized speech is judged as the most natural one among all the methods, despite the variation from different sentences and different subjects. The synthesis system trained in manual alignment gets higher rating than the other three types of synthesis systems trained in forced alignments. Paired-sample t-test between all the 5 systems are performed, and the analysis shows no significant differences between the groups {Mono_4mix, Full_1mix} at the 5% significance level. From Figure 3, except that the copy-synthesized speech is still judged as the most natural one among all the methods, listeners nearly cannot tell the differences between the four synthesis systems. Paired-sample t-test shows no significant differences between the groups {Manual, Mono_4mix}, {Manual, Tri_4mix}, {Mono_4mix, Tri_4mix}, {Mono_4mix, Full_1mix} at the 5% significance level.

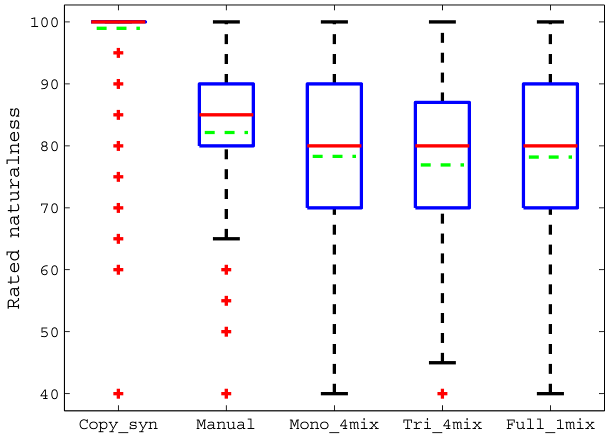

Figure 2: MUSHRA test results on the testing set using manual alignment for 4 different phoneme alignment trained DNNs.

In summary, we can draw the conclusion that the forced alignments are as good as manual alignments for a DNN-based speech synthesis system when the alignment used in training stage and synthesis stage are matched, i.e., from the same alignment approach.

6. Conclusions

In this paper, we studied the impact of phoneme alignment to the performance of DNN-based speech synthesis. We first compared the alignment accuracy of three HMM-based forcing alignment methods with the manual alignment as the reference. Among the three forcing alignment methods, mono-phone model, which is the simplest, is able to achieve similar or even better performance than that of more complex models such as tri-phone and full-context models. We then conducted two groups of speech synthesis experiments to assess their impact on the naturalness of DNN-based speech synthesis. Results show that simple mono-phone HMM-based forced alignment is comparable to manual alignment for DNN speech synthesis performance when the alignment used in the training and synthesis stages are matched.

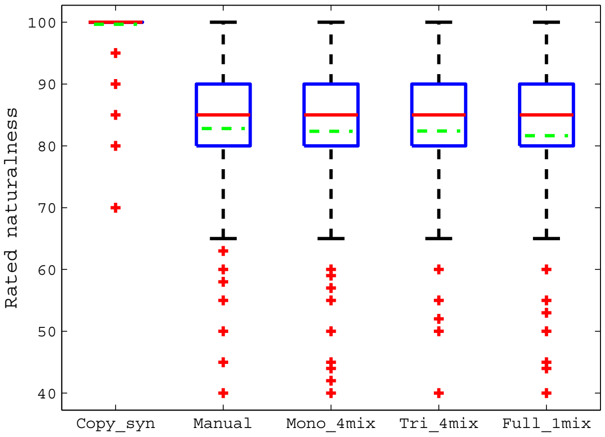

Figure 3: MUSHRA test results on the testing set using their corresponding phoneme alignment for 4 different phoneme alignment trained DNNs.

7. References

[1] H. Zen, K. Tokuda, and A. W. Black, “Statistical parametric speech synthesis,” *Speech Communication*, vol. 51, no. 11, pp. 1039–1064, 2009.

[2] K. Tokuda, Y. Nankaku, T. Toda, H. Zen, J. Yamagishi, and K. Oura, “Speech synthesis based on hidden markov models,” *Proceedings of the IEEE*, vol. 101, no. 5, pp. 1234–1252, 2013.

[3] K. Tokuda, T. Masuko, N. Miyazaki, and T. Kobayashi, “Multi-space probability distribution hmm,” *IEICE TRANSACTIONS on Information and Systems*, vol. 85, no. 3, pp. 455–464, 2002.

[4] T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, “Simultaneous modeling of spectrum, pitch and duration in hmm-based speech synthesis,” in *Proc. Eurospeech*, 1999, pp. 2347–2350.

[5] H. Zen, A. Senior, and M. Schuster, “Statistical parametric speech synthesis using deep neural networks,” in *Proc. ICASSP*. IEEE, 2013, pp. 7962–7966.

[6] Y. Qian, Y. Fan, W. Hu, and F. K. Soong, “On the training aspects of deep neural network (dnn) for parametric tts synthesis,” in *Proc. ICASSP*. IEEE, 2014, pp. 3829–3833.

[7] Z. Wu, C. Valentini-Botinhao, O. Watts, and S. King, “Deep neural networks employing multi-

task learning and stacked bottleneck features for speech synthesis," in *Proc. ICASSP*. IEEE, 2015, pp. 4460–4464.

[8] H. Zen and A. Senior, "Deep mixture density networks for acoustic modeling in statistical parametric speech synthesis," in *Proc. ICASSP*. IEEE, 2014, pp. 3844–3848.

[9] Y. Fan, Y. Qian, F.-L. Xie, and F. K. Soong, "Tts synthesis with bidirectional lstm based recurrent neural networks." in *Proc. INTERSPEECH*, 2014, pp. 1964–1968.

[10] Z. Wu and S. King, "Investigating gated recurrent neural networks for speech synthesis," in *Proc. ICASSP*, 2016.

[11] K. Hashimoto, K. Oura, Y. Nankaku, and K. Tokuda, "The effect of neural networks in statistical parametric speech synthesis," in *Proc. ICASSP*. IEEE, 2015, pp. 4455–4459.

[12] O. Watts, G. E. Henter, T. Merritt, Z. Wu, and S. King, "From hmms to dnns: where do the improvements come from?" in *Proc. ICASSP*, 2016.

[13] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *Signal Processing Magazine, IEEE*, vol. 29, no. 6, pp. 82–97, 2012.

[14] M. Chu, Y. Chen, Y. Zhao, Y. Li, and F. Soong, "A study on how human annotations benefit the tts voice," *Blizzard Challenge Workshop*, 2006.

[15] P. Cosi, D. Falavigna, and M. Omologo, "A preliminary statistical evaluation of manual and automatic segmentation discrepancies." in *Proc. Eurospeech*, 1991.

[16] A. Ljolje and M. D. Riley, "Automatic segmentation of speech for tts," in *Proc. Eurospeech*, 1993.

[17] D. T. Toledano, L. A. H. ndez Gómez, and L. V. Grande, "Automatic phonetic segmentation," *IEEE Trans. Speech and Audio Processing*, vol. 11, pp. 617–625, 2003.

[18] S. S. Park and N. S. Kim, "On using multiple models for automatic speech segmentation," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 15, no. 8, pp. 2202–2212, 2007.

[19] S. Brognaux, S. Roekhaut, T. Drugman, and R. Beaufort, "Automatic phone alignment," in *Advances in Natural Language Processing*. Springer, 2012, pp. 300–311.

[20] J. Yuan, N. Ryant, M. Liberman, A. Stolcke, V. Mitra, and W. Wang, "Automatic phonetic segmentation using boundary models." in *Proc. INTERSPEECH*, 2013.

[21] Y.-J. Wu, H. Kawai, J. Ni, and R.-H. Wang, "Minimum segmentation error based discriminative training for speech synthesis application," in *Proc. ICASSP*, vol. 1. IEEE, 2004, pp. I–629.

[22] A. W. Black and J. Kominek, "Optimizing segment label boundaries for statistical speech synthesis." in *Proc. ICASSP*, 2009, pp. 3785–3788.

[23] J. Adell, A. Bonafonte, J. A. Gómez, and M. J. Castro, "Comparative study of automatic phone segmentation methods for tts." in *Proc. ICASSP*, 2005.

[24] S. Young, G. Evermann, M. Gales, T. Hain, D. Kershaw, X. Liu, G. Moore, J. Odell, D. Ollason, D. Povey *et al.*, "The htk book (v3. 4)," *Cambridge University*, 2006.

[25] H. Zen, T. Nose, J. Yamagishi, S. Sako, T. Masuko, A. W. Black, and K. Tokuda, "The hmm-based speech synthesis system (hts) version 2.0." in *Speech Synthesis Workshop*, 2007.

[26] J. Bergstra, O. Breuleux, F. Bastien, P. Lamblin, R. Pascanu, G. Desjardins, J. Turian, D. Warde-Farley, and Y. Bengio, "Theano: a cpu and gpu math expression compiler," in *Proc. SciPy*, vol. 4. Austin, TX, 2010, p. 3.

[27] A. Ljolje, J. Hirschberg, and J. P. van Santen, "Automatic speech segmentation for concatenative inventory selection," in *Progress in speech synthesis*. Springer, 1997, pp. 305–311.

[28] B. Series, "Method for the subjective assessment of intermediate quality level of audio systems," *International Telecommunication Union Radiocommunication Assembly*, 2014.