Multi-Language Multi-Speaker Acoustic Modeling for LSTM-RNN based Statistical Parametric Speech Synthesis

Bo Li, Heiga Zen

Google Inc.
{boboli,heigazen}@google.com

Abstract

Building text-to-speech (TTS) systems requires large amounts of high quality speech recordings and annotations, which is a challenge to collect especially considering the variation in spoken languages around the world. Acoustic modeling techniques that could utilize inhomogeneous data are hence important as they allow us to pool more data for training. This paper presents a long short-term memory (LSTM) recurrent neural network (RNN) based statistical parametric speech synthesis system that uses data from multiple languages and speakers. It models language variation through cluster adaptive training and speaker variation with speaker dependent output layers. Experimental results have shown that the proposed multilingual TTS system can synthesize speech in multiple languages from a single model while maintaining naturalness. Furthermore, it can be adapted to new languages with only a small amount of data.

Index Terms: statistical parametric speech synthesis, multilingual acoustic modeling, long short-term memory, recurrent neural networks

1. Introduction

Deep neural networks (DNNs) have been widely adopted in various applications in machine learning, including text-to-speech (TTS) systems. Zen *et al.* [1] first introduced feed forward DNNs to statistical parametric speech synthesis [2] and showed that DNNs could achieve better naturalness than the conventional hidden Markov model (HMM) based systems [3] even with the same number of parameters. Later, long short-term memory (LSTM) based recurrent neural networks (RNNs) were adopted to model the inherent temporal and long-term dependencies in speech signals [4, 5], which shifted the parametric TTS from frame-based modeling to more desired sequence-based modeling. Unlike HMMs, neural network models can handle very large amount of training data, which they commonly require. However, for TTS systems, high quality speech recordings and annotations are required, which becomes a bottleneck for neural network based statistical parametric speech synthesis.

Human speech contains rich information besides the linguistic meaning, such as individual speaker characteristics and emotional states. Being able to analyze, understand and model variation is of crucial importance if it occurs within a dataset. In conventional speech processing systems, acoustic factorization [6] has provided a good foundation for modeling inhomogeneous variation. It represents each affecting factor with separate transforms and then builds a canonical model set given the combined transform for all the factors. Speaker and language are the two primary factors that influence speech generation. To model them, a speaker and language factorization (SLF) framework was proposed and justified for HMM-based statistical parametric speech synthesis [7]. It represents the speaker characteristics with the constrained maximum likelihood linear regression (CMLLR) transforms [8] and models the language information through cluster adaptive training [9]. With this factorization, speech data from different languages and speakers could be utilized to build a single TTS system. Even if only a single language is required, for a limited data scenario, the increased amount of training data for acoustic modeling obtained by using speech data from multiple speakers in different languages would also be beneficial. More importantly, if the amount of data from a new language is limited, the synthesis system can be adapted to the new language by estimating only the corresponding language and speaker transforms, which have far fewer parameters and can be more reliably trained.

This paper explores the building of TTS systems with speech data from different languages and speakers. Inspired from the SLF framework, the proposed system models language variation through cluster adaptive training and speaker variation with speaker dependent output layers. The rest of the paper is organized as follows: section 2 explains the proposed multi-language multi-speaker (MLMS) TTS system in detail; section 3 and section 4 experimentally justify the proposed system through both objective and subjective evaluations. Concluding remarks are presented in the last section.

2. Multi-Language Multi-Speaker Acoustic Modeling

The proposed multi-language multi-speaker (MLMS) acoustic modeling system adopts cluster adaptive training (CAT) [10] for modeling the language variation and speaker dependent output layers [11] for the speaker variation. The LSTM model structure used for building up the MLMS system is similar to the unidirectional LSTM-based TTS system developed in [5], which consists of an input projection layer with rectified linear activation function (ReLU) [12], several LSTM [13] layers and an RNN output layer [5].

In the MLMS system, given a sequence of words from language u, a language dependent text analysis module is first run to extract a sequence of universal linguistic feature vectors. One way to define a universal linguistic feature set is to map each language to a canonical representation, such as those based on the international phonetic alphabet (IPA) [14]. However, for simplicity, the union of the linguistic feature sets of all the languages is used in the MLMS system. In this way, for each language, the same text analysis module that is used for building single language, single speaker TTS systems can be used directly. After that, zero padding is used for feature dimensions that are not available in the current language, which extends those language dependent linguistic feature vectors to the universal represen-

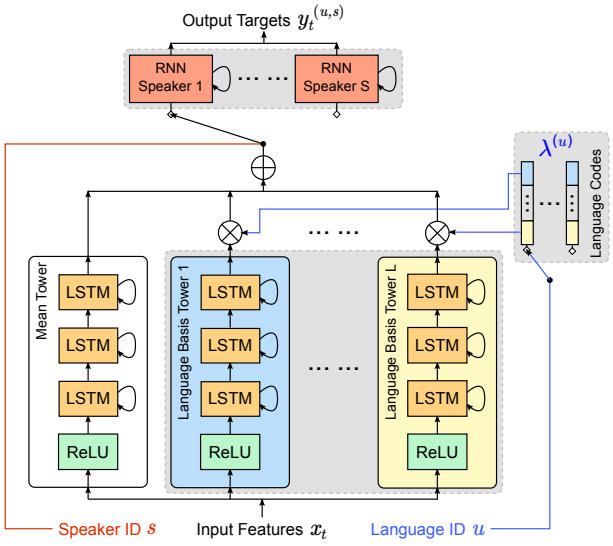

Figure 1: *Architecture for the LSTM-based multi-language multi-speaker TTS system. It consists of a mean tower, L language basis towers and S speaker dependent RNN output layers.*

tation for all the languages. A duration model is then used to convert them to a sequence of frame-level linguistic feature vectors: $\{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_t, \cdots, \boldsymbol{x}_T\}$. In this study, the oracle duration information was used for simplicity. For each frame at time step t, the feature vector $\boldsymbol{x}_t$ together with a language ID u and a speaker ID s is passed through the LSTM-based MLMS acoustic model (Figure 1) to output a vocoder parameter feature vector $\boldsymbol{y}_t^{(u,s)}$ for the desired language u and speaker s, which is then forwarded to a vocoder [15] to synthesize the final waveform signal. As depicted in Figure 1, the LSTM-based MLMS acoustic model consists of four major components:

1) mean tower $\mathcal{M}^{\texttt{mean}}$, a sub-network of any type, which captures the shared knowledge across different training languages.

2) language basis towers $\mathcal{M}_l^{\texttt{lang}}$ for $l \in \{1, \cdots, L\}$, a set of sub-networks trained to capture different variation. For this study, they were optimized to build up a language space that models the variation of all the training languages. L is the dimension of the language space.

3) language code vector $\boldsymbol{\lambda}^{(u)}$ for each of the training languages $u \in \{1, \cdots, U\}$. The total number of training languages, U, is usually larger than the dimension of the language space, L, which forces the clustering of the given set of training languages and encourages information sharing.

4) speaker dependent RNN output layer $\mathcal{M}_s^{\texttt{spkr}}$ one for each training speaker $s \in \{1, \cdots, S\}$.

The vocoder parameters generated by the MLMS system for language u and speaker s, $\boldsymbol{y}_t^{(u,s)}$, can be derived as follows:

$$\boldsymbol{h}_t^{(u)} = \mathcal{M}^{\texttt{mean}}(\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_t\}) + \sum_{l=1}^{L} \lambda_l^{(u)} \mathcal{M}_l^{\texttt{lang}}(\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_t\}) \quad (1)$$

$$\boldsymbol{y}_t^{(u,s)} = \mathcal{M}_s^{\texttt{spkr}}(\boldsymbol{h}_t^{(u)}, \boldsymbol{y}_{t-1}^{(u,s)}) \quad (2)$$

The language variation is modeled using a basis representation (Equation 1). The language mean tower, $\mathcal{M}^{\texttt{mean}}$, sets up the origin of the language space and each language basis tower, $\mathcal{M}_l^{\texttt{lang}}$, learns one potential direction of variation away from the origin. Theoretically, there is no constraint on the model structure for each tower. Each of them could have its own structure. However, in this paper, the same tower structure as the mean tower (Figure 1) is used for all the language basis towers. Specifically, within each tower, the input feature vector is first passed through a ReLU projection layer, which transforms the usually sparse input vector into a dense representation that is more suitable for the following neural network layers [4]. After that, 3 LSTM layers are used to model the complex mapping from linguistic features to vocoder parameters. With the activations generated from each of the towers, a summation of the mean tower with a weight of 1.0 and the weighted combination of the language basis towers with weights from the language code vector $\boldsymbol{\lambda}^{(u)}$ produces the intermediate language dependent hidden activation $\boldsymbol{h}_t^{(u)}$.

The language code vector (Figure 1), $\boldsymbol{\lambda}^{(u)}$, locates the language u in the L dimensional space and converts the universal input linguistic feature vector $\boldsymbol{x}_t$ to language dependent hidden activations $\boldsymbol{h}_t^{(u)}$. To find a good set of bases, eigen analysis is commonly adopted in conventional basis decomposition methods to extract independent directions [16]. However, as the basis models are stacks of neural network layers, it is challenging to conduct eigen analysis while maintaining the modeling functionality. Therefore, a data-driven approach to learn bases directly from training is used. With the basis decomposition, information about each language is encoded in an L dimensional vector $\boldsymbol{\lambda}^{(u)}$, which is usually very small compared to the number of parameters in the bases. Adding a new language requires simply learning only L numbers. This is preferable when we have very limited data. However, when we gradually accumulate an adequate amount of training data, updating only the basis may limit the adaptation capability. The mean tower of the MLMS system is hence also adapted to shift the language space towards the target language when a reasonable amount of data for the target language is available, which is similar to adding a new tree in [7].

The speaker variation is modeled through speaker dependent output layers. Given a speaker ID s, the MLMS system will retrieve the corresponding output layer for that speaker and use it for converting the language dependent hidden activation $\boldsymbol{h}_t^{(u)}$ to the corresponding output vocoder parameters for the specific language-speaker pair (u, s). Similarly, the basis decomposition approach could also be adopted for speaker factorization [17]. However, in consideration of the relatively small number of parameters in the RNN output layer, where the estimation is usually robust with a proper amount of data, factor dependent layers are chosen for this case. Moreover, adapting the whole output layer gives more flexibility compared to adapting only a few basis coefficients in modeling the variation across speakers.

During training, the whole system is randomly initialized and jointly trained to minimize the mean square error between the predicted and target vocoder parameters. For a language and speaker pair (u, s), the mean tower $\mathcal{M}^{\texttt{mean}}$, all the language basis towers $\mathcal{M}_l^{\texttt{lang}}$, the corresponding language code vector $\boldsymbol{\lambda}^{(u)}$ and the specific speaker dependent RNN output layer $\mathcal{M}_s^{\texttt{spkr}}$ will be updated. The other language code vectors and speaker dependent RNN output layers will remain unchanged unless the corresponding data samples are provided. For the estimation of language basis and language code vector, we could do an iterative estimation; however, in this paper, a direct optimization without explicitly scheduling the estimation order was sufficient.

3. Experimental Details

The multilingual speech database used in this paper was constructed by simply pooling data from different languages for existing single language, single speaker TTS systems. Six training languages were used: North American (US) English, British (UK) English, French, Italian, German and Spanish. There was only one female professional speaker in each language, except for US English and UK English, where one additional male professional speaker's data existed so these were added also to bring more speaker variation during training. Two testing languages, Polish and Brazilian (BR) Portuguese with one single female professional speaker each, were used as limited resource language adaptation targets. Detailed information about the data is tabulated in Table 1.

Stage	Language	Speaker	Train	Dev
Training	US English	female	35,493	100
		male	7,652	
	UK English	female	21,403	
		male	3,587	
	French	female	29,924	
	Italian	female	29,897	
	German	female	24,535	
	Spanish	female	19,872	
Testing	Polish	female	900	
	BR Portuguese	female	900	

Table 1: *Number of utterances used in each language for training and development.*

From the speech data and its associated transcriptions, phonetic alignments were automatically generated using an HMM-based aligner, which was trained in a bootstrap manner. Phoneme-level linguistic features for each language were pooled together to form a universal phoneme-level linguistic feature set (*e.g.* phoneme identities, stress marks, the number of syllables in a word, position of the current syllable in a phrase). Then the universal phoneme-level linguistic features, 3 numerical features for coarse-coded position of the current frame in the current phoneme and 1 numerical feature for duration of the current segment were used to form frame-level linguistic features. Detailed frontend processing could be found in [1, 5].

The speech analysis conditions were similar to those used for Nitech-HTS 2005 [18] system. The speech data was downsampled from 48kHz to 22.05kHz, then 40 mel-cepstral coefficients [19], logarithmic fundamental frequency (log F_0) values, and 7-band aperiodicity [18] were extracted every 5 ms. The output features of the LSTM-based MLMS acoustic model were vocoder parameters consisting of 40 mel-cepstral coefficients, log F_0 value, and 7 band aperiodicities. To model log F_0 sequences, the continuous F_0 with explicit voicing modeling approach [20] was used; voiced/unvoiced binary value was added to the output features and log F_0 values in unvoiced frames were interpolated. Benefiting from LSTM's long-term dependency modeling capability and RNN output layer's smoothing effect, no dynamic features were used.

Both the input and output features were normalized to have zero-mean and unit-variance. All the model weight parameters were randomly initialized (no pretraining was performed) and then updated to minimize the mean squared error between the target and predicted output features. A distributed CPU implementation of mini-batch ASGD-based [21] truncated back propagation through time (BPTT) [22] algorithm was used. The same training configuration used for building single-language single-speaker systems [5] was adopted without additional tunning for this new structure. Training was continued until the mean squared error over the development set converged.

At the synthesis stage, vocoder parameters were predicted from linguistic features using the trained networks and directly passed to the latter vocoding step. Spectral enhancement based on post-filtering in the cepstral domain was applied to improve the naturalness of the synthesized speech. Natural speech durations were used in this study. From the vocoder parameters, speech waveforms were synthesized using the Vocaine vocoder [15].

To subjectively evaluate the performance of the systems, preference tests were conducted. 100 utterances not included in the training data were used for evaluation. For each language, only the native speakers using headphones were allowed to take part in the listening tests. One subject could evaluate a maximum of 30 pairs and each pair was evaluated by eight subjects in the preference test. Subjects were asked to choose the preferred sample from each pair of speech samples.

4. Results

4.1. Building a Multi-Language Multi-Speaker System

This experiment aims to justify the capability of the proposed LSTM-based MLMS system in modeling the large variations across languages and speakers. Six basis towers and one mean tower were used to model the language space. In each tower, the input linguistic feature vector was first projected down to a 256 dimensional dense representation by a ReLU layer. After that, 3 LSTM layers were used to model the long-term temporal dependencies. Each LSTM had 256 memory cells and an output projection layer which condensed the 256 dimensional LSTM output vector into a lower dimensionality of 128. The whole system output 49 dimensional vocoder parameter vectors. The speaker dependent RNN output layer hence contained a forward matrix with the dimensionality of 128×49, a recurrent matrix of the size 49×49 and a 49 dimensional bias vector. For comparison purpose, we also trained the single language single speaker LSTM baseline models, one for each language-speaker pair. These baseline models have the same model structure, namely one 256-dimensional ReLU projection layer, three LSTMs each with 256 memory cells and a 128-dimension output projection, and one RNN output layer with 49-dimension output.

During training, all the towers and speaker dependent RNN layers were randomly initialized. For the code vector of each language, three different setups were tried: 1) `rand` - randomly initialized and updated in training; 2) `1-hot(init)`: initialized with the 1-hot vector representation for each language and updated in training; 3) `1-hot(fixed)`: initialized with 1-hot vectors and kept constant. As there are only two languages having male speakers and they also have much less data compared to female speakers, evaluations were only conducted on the female speaker for each language. From the final mean square error on the development set of each language shown in Figure 2, the simple random initialization worked the best among the three different language code vector training strategies. Hence, only the model trained with randomly initialized language code vectors was used for the following investigations.

In Figure 2, the proposed MLMS system has slightly worse objective scores compared to the single language, single speaker baseline systems. Subjective preference listening tests were also

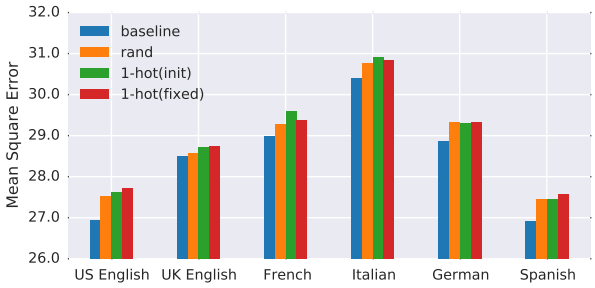

Figure 2: *Mean square error on dev set of baseline single language, single speaker systems and systems using three different language code training strategies respectively.*

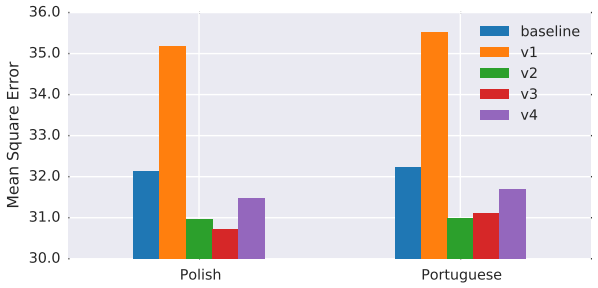

Figure 3: *Mean square error on dev set of different systems adapting to Polish and BR Portuguese with limited data.*

conducted to further justify the proposed MLMS system. The results are tabulated in Table 2. In the preference test, the single language, single speaker baseline models were referred to as system `A`, while the proposed LSTM-based MLMS system was referred to as system `B`. Except for UK English, building a single system from multiple languages and speakers using the proposed LSTM-based MLMS achieved the same naturalness as building a set of models one per language and speaker pair.

Language	**Subjective Preference (%)**		
	`A`	`B`	`Neutral`
US English	11.5	11.0	77.5
UK English	**25.9**	13.8	60.4
French	12.6	13.9	73.5
Italian	13.1	17.5	69.4
German	18.3	18.3	63.5
Spanish	13.9	11.6	74.5

Table 2: *Subjective preference scores (%) on training languages between the baseline single language single speaker LSTM systems (`A`) and the proposed LSTM-based MLMS system (`B`). The systems which achieved significantly better preference at $p < 0.01$ level are in the bold font.*

4.2. Adaptation to New Languages

In this set of experiments, the effectiveness of adapting the proposed LSTM-based MLMS system to new languages, namely, Polish and BR Portuguese, were verified with limited training data. To adapt to the target speaker, we needed to update the RNN output layer; while for the language adaptation, we could either only update the language code or the mean tower, or update them alternatively, or jointly. We hence experimented with the following set of experiments:

`v1` : update only language code $\boldsymbol{\lambda}^{(u)}$;

`v2` : update language code $\boldsymbol{\lambda}^{(u)}$ and mean tower $\mathcal{M}^{\texttt{mean}}$ jointly;

`v3` : start from `v1`, update mean tower $\mathcal{M}^{\texttt{mean}}$ alone;

`v4` : start from `v3`, joint update language code $\boldsymbol{\lambda}^{(u)}$ and mean tower $\mathcal{M}^{\texttt{mean}}$.

The development set mean square error of the above systems and baseline models are depicted in Figure 3. Other than `v1`, which has very constrained adaptability, all the other systems yield lower mean square error than directly building a system from the limited data. It clearly suggests the benefit of using the proposed LSTM-based MLMS system for new languages with limited training data (900 utterances per language, *cf.* Table 1). For Polish, `v3` has the lowest mean square error; while for BR Portuguese, `v2` works slightly better than `v3`.

To further verify the gains, subjective preference tests among the baseline system, `v2` and `v3` were conducted. The results are summarized in Table 3. For Polish, `v3` was statistically significant better than the baseline ($p < 0.01$). For BR Portuguese, `v2` and `v3` were both better than the baseline ($p < 0.01$) and there was no statistically significant difference between them.

Language	**Subjective Preference (%)**			
	`A`	`v2`	`v3`	`Neutral`
Polish	31.1	38.3	-	30.6
	30.0	-	**41.5**	28.5
	-	33.1	39.8	27.1
BR Portuguese	14.4	**23.4**	-	62.3
	13.3	-	**25.6**	61.1
	-	10.4	10.4	79.3

Table 3: *Subjective preference scores (%) for adaptation to new languages with limited data among the baseline single language, single speaker LSTM systems (`A`) and the two adaptation strategies for the proposed LSTM-based MLMS system (`v2` and `v3`). The systems which achieved significantly better preference at $p < 0.01$ level are in the bold font.*

5. Conclusions

This paper presented an LSTM-based multi-language multi-speaker (MLMS) statistical parametric speech synthesis system to utilize inhomogeneous data. It models the language variation through cluster adaptive training where language basis towers and language code vectors are jointly learned during training. The speaker variation is captured through speaker dependent RNN output layers. Experimental results on six languages have shown that the proposed MLMS system achieves similar performance on the training languages and speakers, compared to conventional language and speaker dependent models. Moreover, adaptation of the LSTM-based MLMS system to new languages with limited training data yields much better performance in both objective and subjective evaluations than building models from scratch. Future work includes exploring other adaptation techniques such as speaker code and better optimization techniques especially when training with more languages which we have found to be more challenging.

6. References

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

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

[3] 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.

[4] 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.

[5] H. Zen and H. Sak, "Unidirectional long short-term memory recurrent neural network with recurrent output layer for low-latency speech synthesis," in *Proc. ICASSP*. IEEE, 2015, pp. 4470–4474.

[6] M. Gales, "Acoustic factorisation," in *Proc. ASRU*. IEEE, 2001, pp. 77–80.

[7] H. Zen, N. Braunschweiler, S. Buchholz, M. Gales, K. Knill, S. Krstulović, and J. Latorre, "Statistical parametric speech synthesis based on speaker and language factorization," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 20, no. 6, pp. 1713–1724, 2012.

[8] M. Gales, "Maximum likelihood linear transformations for HMM-based speech recognition," *Computer speech & language*, vol. 12, no. 2, pp. 75–98, 1998.

[9] M. Gales, "Cluster adaptive training of hidden Markov models," *Speech and Audio Processing, IEEE Transactions on*, vol. 8, no. 4, pp. 417–428, 2000.

[10] T. Tan, Y. Qian, M. Yin, Y. Zhuang, and K. Yu, "Cluster adaptive training for deep neural network," in *Proc. ICASSP*. IEEE, 2015, pp. 4325–4329.

[11] Y. Fan, Y. Qian, F. K. Soong, and L. He, "Multi-speaker modeling and speaker adaptation for DNN-based TTS synthesis," in *Proc. ICASSP*. IEEE, 2015, pp. 4475–4479.

[12] M. D. Zeiler, M. Ranzato, R. Monga, M. Mao, K. Yang, Q. V. Le, P. Nguyen, A. Senior, V. Vanhoucke, J. Dean *et al.*, "On rectified linear units for speech processing," in *Proc. ICASSP*. IEEE, 2013, pp. 3517–3521.

[13] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[14] International Phonetic Association, *Handbook of the International Phonetic Association: A guide to the use of the International Phonetic Alphabet*. Cambridge University Press, 1999.

[15] Y. Agiomyrgiannakis, "Vocaine the vocoder and applications in speech synthesis," in *Proc. ICASSP*. IEEE, 2015, pp. 4230–4234.

[16] R. Kuhn, P. Nguyen, J. C. Junqua, L. Goldwasser, N. Niedzielski, S. Fincke, K. Field, and M. Contolini, "Eigenvoices for speaker adaptation," in *ICSLP*, vol. 98, 1998, pp. 1774–1777.

[17] V. Wan, J. Latorre, K. Yanagisawa, N. Braunschweiler, L. Chen, M. Gales, and M. Akamine, "Building HMM-TTS voices on diverse data," *Selected Topics in Signal Processing, IEEE Journal of*, vol. 8, no. 2, pp. 296–306, 2014.

[18] H. Zen, T. Tomoki, M. Nakamura, and K. Tokuda, "Details of the Nitech HMM-based speech synthesis system for the Blizzard Challenge 2005," *IEICE transactions on information and systems*, vol. 90, no. 1, pp. 325–333, 2007.

[19] T. Fukada, K. Tokuda, T. Kobayashi, and S. Imai, "An adaptive algorithm for mel-cepstral analysis of speech," in *Proc. ICASSP*, vol. 1. IEEE, 1992, pp. 137–140.

[20] K. Yu and S. Young, "Continuous F0 modeling for HMM based statistical parametric speech synthesis," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 19, no. 5, pp. 1071–1079, 2011.

[21] J. Dean, G. Corrado, R. Monga, K. Chen, M. Devin, M. Mao, A. Senior, P. Tucker, K. Yang, Q. V. Le *et al.*, "Large scale distributed deep networks," in *Advances in Neural Information Processing Systems*, 2012, pp. 1223–1231.

[22] R. Williams and J. Peng, "An efficient gradient-based algorithm for on-line training of recurrent network trajectories," *Neural computation*, vol. 2, no. 4, pp. 490–501, 1990.