INTERSPEECH 2016
September 8–12, 2016, San Francisco, USA

Speaker Representations for Speaker Adaptation in Multiple Speakers' BLSTM-RNN-based Speech Synthesis

Yi Zhao[1], *Daisuke Saito*[2], *Nobuaki Minematsu*[1]

[1]Graduate School of Engineering, The University of Tokyo, Japan
[2]Graduate School of Information Science and Technology, The University of Tokyo, Japan
{zhaoyi, dsk_saito, mine}@gavo.t.u-tokyo.ac.jp

Abstract

Training a high quality acoustic model with a limited database and synthesizing a new speaker's voice with a few utterances have been hot topics in deep neural network (DNN) based statistical parametric speech synthesis (SPSS). To solve these problems, we built a unified framework for speaker adaptive training as well as speaker adaptation on Bidirectional Long Short-Term Memory with Recurrent Neural Network (BLSTM-RNN) acoustic model. In this paper, we mainly focus on speaker identity control at the input layer of our framework. We have investigated i-vector and speaker code as different speaker representations when used in an augmented input vector, and also propose two approaches to estimate a new speaker's code. Experimental results show that the speaker representations input to the first layer of acoustic model can effectively control speaker identity during speaker adaptive training, thus improving the synthesized speech quality of speakers included in training phase. For speaker adaptation, speaker code estimated from MFCCs can achieve higher preference than other speaker representations.
Index Terms: speaker adaptive training, speaker adaptation, speaker code, i-vector, RNN-BLSTM

1. Introduction

Compared with unit-selection and concatenative approaches, Statistical Parametric Speech Synthesis (SPSS) is preferred because it can generate natural sounding synthetic speech with rather small corpus and vary speaker identity and speaking styles flexibly. Recently, Deep Neural Network (DNN) has significantly advanced the performance of SPSS. However, it still suffers from necessity of a large recording corpus of one speaker to train a high quality acoustic model [1, 2, 3, 4, 5, 6]. Meanwhile, significant efforts have been made to generate a new speaker's voices with only a few utterances [5, 6, 7, 8].

Speaker adaptive training is one of the most effective approaches to train a high quality acoustic model with a limited database [5, 6, 9, 10]. In speaker adaptive training, acoustic model is jointly trained utilizing multiple speakers' data. For DNN, it has been experimentally proved that the shared hidden layers can benefit synthesized speech of each speaker from the knowledge of others [5, 6]. Speaker adaptation has been developed for generating an arbitrary speaker's voice with minimum adaptation data, and the adaptation process is usually performed on a well trained acoustic model [7, 9, 11]. However, both speaker adaptive training and speaker adaptation techniques need to control speaker identity precisely.

Generally, there are three ways to control the speaker identity in a DNN-based acoustic model. The first way is to control the speaker identity at the input layer, such as adding speaker information as auxiliary input features [7, 8]. The second one is to control the speaker identity with specially designed hidden layers, such as learning hidden unit contribution (LHUC) [7]. And the last one is to control the speaker identity near the output layer space, such as speaker dependent regression or feature space transformation [5, 6, 7].

In our work, we mainly focus on speaker identity control at the input layer. Augmented speaker identity vectors are prepared independently from linguistic features and these vectors can distinguish different speakers very well even if the speakers share the same linguistic labels. In addition to i-vector, speaker code is another speaker representation which has been widely used in DNN-based speaker adaptation in automatic speech recognition [12, 13, 14, 15]. In ASR, a new speaker's code is usually estimated from the training speakers in a backpropogation manner [14, 15]. In SPSS, the estimation of a new speaker's code remains to be solved.

In this paper, we conduct an exploring and comparative study on the controllability of different speaker identity representations when they are used in augmented inputs for speaker adaptive training and speaker adaptation in the same framework. Instead of DNN, Bidirectional Long Short-Term Memory with Recurrent Neural Network (BLSTM-RNN) acoustic model is employed due to its strong capability of learning long-range dynamics of speech as well as variation in speaker identity. We first briefly describe the framework of multiple speaker BLSTM-RNN-based speech synthesis. Then we introduce different speaker identity representations including i-vector and speaker code. We also propose two approaches especially to estimate a new speaker's code. Analytical experiments are done to examine the performance of each speaker representation in both speaker adaptive training and speaker adaptation.

2. Framework for multi-speaker speech synthesis and adaptation

In this section, we mainly introduce our experimental framework used for speaker adaptive training and speaker adaptation. In order to examine the performance of different speaker representations when they are input to the first layer, only conventional BLSTM-RNN acoustic model is adopted, with no specially designed layer or feature space mapping as post-processing. The schematic diagram of our framework is shown in Fig.1.

In this work, we utilize a hybrid network structure which includes both feedforward and BLSTM-RNN layers in acoustic model. The feedforward layer, trained with a back-propagation learning algorithm [16], is widely used in many practical applications. But the assumption of sample independence results in limited ability of modeling context information as well as acoustic signals [3, 17]. Bidirectional RNN can access both the preceding and succeeding input contexts with two separate hidden layers. An LSTM architecture, can overcome the gradient

http://dx.doi.org/10.21437/Interspeech.2016-506

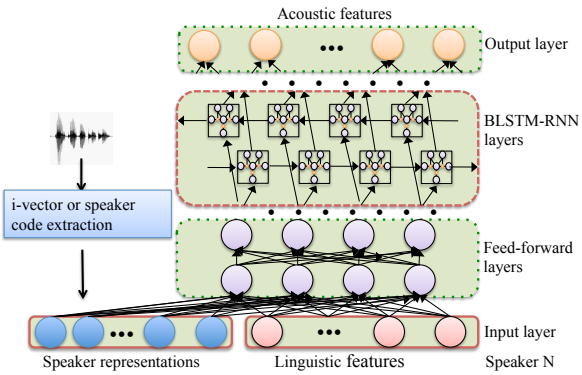

Figure 1: *BLSTM-RNN based multi-speaker speech synthesis model*

vanishing problem that prevents RNN from modeling long-span relations in both linguistic and acoustic domains. Deep bidirectional LSTM-RNN is able to build up progressively higher level representations of input data, which is a crucial factor of the recent success of hybrid systems [17].

Different from the conventional individual speaker's synthesis, the network is trained with multiple speakers' data, and it takes linguistic features as input as well as speaker-specific features, such as speaker code or i-vector. With augmented speaker identity representations, inputs with the same linguistic content but spoken by different speakers can be distinguished. Different from Fan's work in [5], not only hidden layers but also the output layer is shared across all the speakers.

During the speaker adaptive training phase, it is very crucial to train the network for all the speakers simultaneously, which means that each batch should consider the data from all the speakers during the stochastic gradient decent (SGD) procedure, and training data also needs to be shuffled across all the speakers. Since BLSTM-RNN can take use of both past and future information, it can capture the dynamics in speeches as well as speaker identity.

In synthesis, the speeches of any speaker who had ever joined in speaker adaptive training can be synthesized through the well-trained multi-speaker model with the help of speaker specific representation. In speaker adaptation phase, a new speaker's representation is firstly estimated, then appended to linguistic features. The well-trained multi-speaker model is updated with the new speaker's data. The error of new training/adaptation samples are back-propagated to the whole network. The new speaker's speech can be generated with the well-updated model.

3. Speaker identity representations

This section mainly introduces different speaker identity representations including i-vector and speaker code. After that, we propose two methods for speaker code estimation.

3.1. I-vector

I-vector has been widely used in both speaker and speech recognition. It is a low-dimensional vector, the cosine distance between two different i-vectors represents the difference of two speakers: the smaller the distance is, the closer the speakers are, and vice versa. According to [18], by factor analysis, a speaker's supervector M is approximated as

$$\mathbf{M} \approx \mathbf{m} + \mathbf{T}\mathbf{w}, \qquad (\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})), \tag{1}$$

where $\mathbf{m}$ denotes the speaker-independent supervector, which can be extracted easily from the universal background model (UBM), often obtained as GMM. $\mathbf{T}$ is a total variability matrix and $\mathbf{w}$ is a weight vector, so called i-vector.

Since the robust estimation of the total variability matrix $\mathbf{T}$ requires a large amount of data, additional data need to be collected while the number of speakers are limited.

3.2. Speaker Code

Speaker code has achieved promising results in speaker adaptation [14] of automatic speech recognition task. As described in [12], if there are K speakers' corpora for model training, we can simply use $z_c = (z_{1,c}, z_{2,c}, \cdots, z_{k,c}, \cdots z_{K,c})$ to represent the c-th speaker's code. $z_{k,c}$ is defined as follows:

$$z_{k,c} = \begin{cases} 1 & (k = c), \\ 0 & (k \neq c). \end{cases} \tag{2}$$

Although there are many works related to estimate a new speaker's code in ASR, but no work has been reported in SPSS area to the best of our knowledge.

3.2.1. Speaker code estimation from i-vector

Usually, a speaker's i-vector is estimated from his/her utterance-based i-vectors. Those utterance-based i-vectors are able to capture the nuances in speaking style of different utterances, even these utterances are spoken by the same speaker. By collecting utterance-based i-vectors from a single speaker, we can estimate the i-vectors' distribution of that speaker, which is assumed to follow the single Gaussian distribution. If we have K speakers and their corresponding K Gaussian distributions, we can define K-mixture GMM with even weights, assuming that prior probability of each speaker is the same:

$$p(\mathbf{x}) = \sum_{k=1}^{K} \frac{1}{K} \mathcal{N}(\mathbf{x}|\mu_\mathbf{k}, \mathbf{\Sigma_k}) \tag{3}$$

In equation (3), $\mathbf{x}$ represents utterance-level i-vectors , $\mu_\mathbf{k}$ and $\mathbf{\Sigma_k}$ are the mean and variance of utterance-based i-vectors of k-th speaker. The speaker code of a new speaker is estimated as a vector that composed of per-mixture posterior possibility of this K mixture GMM model. Given his/her utterance-based i-vector $\mathbf{x_c}$, the k-th component of the speaker code γ_k is calculated as:

$$\gamma_k = p(k|\mathbf{x_c}) = \frac{\frac{1}{K}\mathcal{N}(\mathbf{x_c}|\mu_\mathbf{k}, \mathbf{\Sigma_k})}{\sum_{j=1}^{K} \frac{1}{K}\mathcal{N}(\mathbf{x_c}|\mu_\mathbf{j}, \mathbf{\Sigma_j})} \tag{4}$$

The new speaker code can be represented as:

$$\tilde{z}_c = (\gamma_1, \gamma_2, \cdots, \gamma_K) \tag{5}$$

3.2.2. Speaker code estimation from MFCCs

Mel-frequency cepstral coefficients (MFCCs) include various kinds of information not only linguistic but also non-linguistic such as speaker identity. Since speaker identity was not drastically changed in an utterance, models which are able to capture the information lying on the long time span are preferable. Hence, direct estimation of speaker code from MFCCs sequences utilizing BLSTM-RNN is investigated in this paper. The input of the network is a gender mark of a speaker and his/her MFCCs for each frame, and the output is always the speaker code of that speaker. This network need to be jointly trained with different speakers' parameters. To get a better classification, a softmax output layer is employed.

4. Experimental setup

In our experiments, the data we used is the CMU ARCTIC corpora [19], which contain 7 speakers (5 males, and 2 females) with parallel sentences. For speaker adaptive training, we chose 900 utterances from each of 4 US English speakers (2 males and 2 females) as training dataset, 100 utterances as validation set, and 50 utterances as testing set. The sentences were common among the four speakers. For speaker adaptation, the other three speakers were treated as the new target speakers. For each speaker, we prepared two training sets, one set had 100 utterances and the other had 30 utterances. 10 sentences were used for validation and 50 utterances were used for testing. For both speaker adaptive training and speaker adaptation, transcriptions of testing sets were common and never covered in training or validation set.

All the wav files were converted into 16KHz sampling raw files, windowed by 25ms. The frame shift is 5ms. Linguistic features were extracted and converted to 307 dimensional vectors. State index and frame index were also attached. State-level alignment was done with the help of HTS [20]. In speaker adaptive training with speaker representations, gender mark as well as i-vector or speaker code were appended to each frame of the linguistic features. Acoustic features including 39-dimensional mel-cepstral coefficients, F0 in log-scale, 26-dimensional Band-aperiodicity parameters (BAP), and their delta and delta-delta features were extracted with the help of STRAIGHT [21]. A binary value for voiced/unvoiced decision was also attached. Linear interpolation of F0 was done over unvoiced segments.

The neural network of speaker independent acoustic model had four hidden layers, including 2 feedforward layers and 2 BLSTM-RNN layers, and each of them had 300 nodes. To train the acoustic model, both input and target features were normalized to zero mean and unit variance. The learning rate was set to 1e-5 and the momentum was set to 0.6. The training was stopped if no improvement was observed within the latest 20 iterations. For speaker adaptation, the adaptation data were used to update the well-trained speaker independent model until no improvement was observed within the latest 20 iterations. Implementation of the network training was done with the help of a machine learning library "CURRENNT" [22].

STRAIGHT was employed to synthesize waveforms from predicted acoustic parameters. Before waveform generation, global variances were used with Maximum Likelihood Parameter Generation (MLPG) algorithm to enhance the dynamic properties of synthetic speech. To calculate the global variances, the variance of each sentence's acoustic features was built as a single GMM.

To extract i-vectors, a gender independent 2048-mixture UBM and 30-dimension total variability matrices were trained with the EM algorithm, using NIST SRE corpora (2004, 2006, 2008), Switchboard II Phase 1/2/3, Switchboard Cellular I/II and the CMU ARCTIC corpora [23, 24, 25, 26, 27, 19]. Then the i-vectors of the input speeches were extracted using the UBM and T matrices.

Speaker codes of the 4 US speakers were prepared according to equation (2). To estimate a new speaker's code using i-vector, 4 mixture GMM was trained with utterance-based i-vectors of the 4 US speakers. For speaker code estimation using MFCCs, 13-dimensional MFCCs parameters were extracted. The neural network had 6 layers, including 3 feedforward layers and 3 BLSTM-RNN layers, the set for the number of nodes in each layer is [50, 200, 400, 300, 200, 100]. A softmax output layer and cross entropy objective function were utilized. The learning rate was set to 1e-4 and the momentum was set to 0.5.

5. Evaluation and discussion

Objective measures used in this paper are Mel-cepstral Distortion (MCD) [28], F0 distortion in the root mean squared error (RMSE), BAP distortion and voiced/unvoiced (V/UV) swapping errors.

As for subjective evaluation, we conducted AB preference tests to evaluate the preference in naturalness and ABX preference tests to evaluate similarity. Here, A and B stand for synthesized speech samples generated by two different systems. X represents the target speaker's raw speech. For naturalness test, 20 English speakers were asked to select A or B which is more natural and comfortable. For similarity test, the listeners were asked to select A or B based on their perceptual similarity to X in terms of speaker identity. If no difference is perceived, the listeners were asked to select the other option, that is neutral. In each experimental group, 20 parallel sentences are selected randomly from testing sets of each system.

In this section, SI represents the speaker independent model which is trained by multiple speakers' data but without speaker identity information. $\mathrm{SC}_{(\mathrm{O})}$stands for the original speaker code as showed in equation (2). $\mathrm{SC}_{(\mathrm{I})}$ means speaker code estimated from i-vectors and $\mathrm{SC}_{(\mathrm{M})}$ is for speaker code estimated from MFCCs.

5.1. Evaluations for multi-speaker speech synthesis

We use the 4 US speakers who have joined in speaker adaptive training to evaluate the quality for multi-speaker synthesis. Table 1 shows the average objective evaluation results over female and male speakers respectively. Individual modeling is trained with only the target speaker's data and used as baseline.

From table 1, we can find that SI presents much higher distortion than baseline in all kinds of objective measures . But when i-vector or speaker code is attached to the input feature, it can outperform the baseline in all aspects. For the female speakers, SI+$\mathrm{SC}_{(\mathrm{O})}$ achieves best performance in F0 RMSE and V/UV error rates while SI+i-vector gives the lowest distortion in MCD and BAP. For the male speaker, SI+$\mathrm{SC}_{(\mathrm{O})}$ gives the lowest distortion in MCD, F0 and BAP while SI+i-vector shows the lowest V/UV error rates.

Subjective evaluation results are showed in Fig.2. SI + $\mathrm{SC}_{(\mathrm{O})}$ and SI + i-vector gets much higher preference than the individually modeling while individual modeling is preferred than SI. Among all systems, SI + $\mathrm{SC}_{(\mathrm{O})}$ achieves the most preference in terms of both naturalness and similarity, and we hope it can achieve good performance in speaker adaptation.

The evaluation results not only suggest that the shared hidden layers can help to improve the quality of synthesized speech, but also demonstrate that the speaker representations input to the first layer of BLSTM-RNN can control speaker identity very effectively during speaker adaptive training.

5.2. Evaluations for speaker adaptation

The average objective evaluation results of the new speaker's adaptation are presented in Table 2. Individual speech synthesis systems are trained with the same adaptation data of the new speaker. To illustrate the effect of adaptation, we also trained an individual synthesis system with 900 sentences of the target speaker and the distortions are showed in Table 3.

According to Table 2, distortions of adapted speeches are much lower than the distortion of individual synthesis, which suggests the importance of model initialization. For speaker adaptation, the neural network is updated based on a well-trained multiple speakers' acoustic model, but for individual synthesis it is initialized randomly.

Table 1: *Objective evaluations for multi-speaker's speech synthesis. Individual synthesis is trained by only one speaker's data.*

Speaker	Female				Male			
Experimental Systems	MCD (dB)	F_0 RMSE(Hz)	BAP (dB)	V/UV err(%)	MCD (dB)	F_0 RMSE(Hz)	BAP (dB)	V/UV err(%)
Individual synthesis	5.45	23.68	6.87	4.37	7.15	15.50	3.52	7.21
SI	5.67	31.40	3.89	4.39	5.52	18.36	3.90	7.24
SI+i-vector	**5.04**	20.25	**3.70**	4.02	5.26	13.80	3.29	**6.82**
SI+SC$_{(O)}$	5.06	**19.96**	3.69	**4.01**	**5.24**	**13.40**	**3.28**	7.09

Table 2: *Objective evaluations for new speaker's adaptation. Individual synthesis is trained using the same data of the target speaker.*

Sentences' number	100 sentences				30 sentences			
Experimental Systems	MCD (dB)	F_0 RMSE(Hz)	BAP (dB)	V/UV err(%)	MCD (dB)	F_0 RMSE(Hz)	BAP (dB)	V/UV err(%)
Individual synthesis	8.19	18.16	3.86	14.1	9.06	20.32	4.17	22.6
SI	5.14	20.90	3.31	**7.05**	5.30	22.56	3.35	**7.21**
SI+i-vector	5.19	23.02	3.31	7.41	5.28	21.03	3.32	7.70
SI+SC$_{(I)}$	**5.11**	**19.64**	3.30	7.19	5.28	18.76	3.31	7.35
SI+SC$_{(M)}$	5.16	20.06	**3.29**	7.32	**5.23**	**18.17**	**3.29**	7.77

Table 3: *Objective evaluations for the new speaker's speech synthesis using 900 sentences*

Experimental System	MCD (dB)	F_0 RMSE(Hz)	BAP (dB)	V/UV err(%)
Individual synthesis	7.13	15.06	3.38	7.17

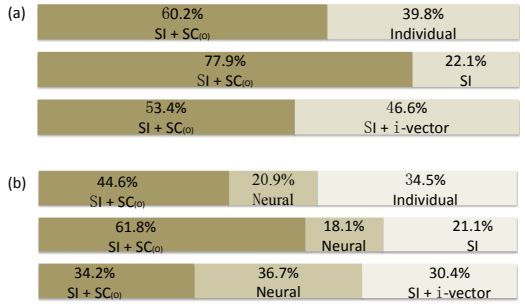

Figure 2: *Subjective evaluations for multi-speaker synthesis in terms of naturalness (a) and similarity (b).*

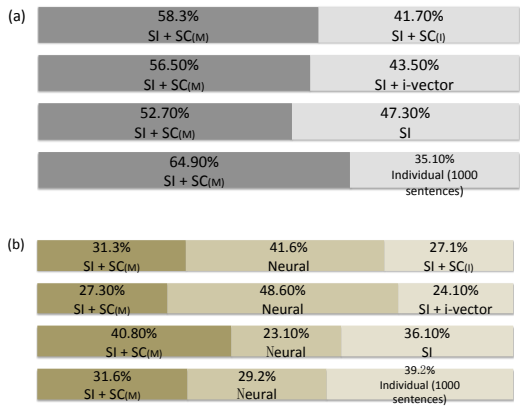

Figure 3: *Subjective evaluations for speaker adaptation in terms of naturalness (a) and similarity (b)*

From Table 2, we can find a very interesting phenomenon that the supervised adaptation based on the speaker independent model can achieve quite small distortion even without i-vector or speaker code, especially in 100 sentences' case. This is quite different from the experimental results of multi-speaker synthesis. It may indicate that speaker information which just input to the first layer may only have very limited guidance in a supervised adaptation way.

By comparing Table 2 and Table 3, it is easy to find that the adaptation system can obtain less MCD distortions than individual speech synthesis using 900 sentences although F0 RMSE error is a little higher. Among all the systems, estimated speaker code can achieve the lowest MCD, RMSE of F0 and BAP distortions, and SI achieves the lowest V/UV error rate.

Another interesting phenomenon is that the speaker code based adaptation always achieves a slight superior to i-vector based adaptation in terms of F0 RMSE. This trend also happens in multi-speaker synthesis. A possible explanation is that the discrete speaker codes are more robust in F0 prediction than i-vectors which are more continuous and have strong relationship to spectral parameters.

Subjective evaluation results are presented in Fig.3. SI + SC$_{(M)}$ obtains better preference than SI+SC$_{(I)}$ in terms of both naturalness and similarity. SI+SC$_{(M)}$ also gets higher evaluation than SI+i-vector and SI. Compared with individual synthesis with 900 sentences, SI + SC$_{(M)}$ achieves more preference in naturalness.

6. Conclusions

In this paper, we mainly investigated the controllability of different speaker representations when they are performed at the input layer of neural networks. Experimental results showed that speaker representations input to the first layer of acoustic model can control speaker identity effectively during speaker adaptive training, but its impact on the supervised adaptation of a new speaker is limited. Further, we will explore to perform speaker identity control at different layers of neural network.

7. Acknowledgement

This work was supported by MEXT KAKENHI Grant Number JP26118002.

8. References

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

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

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

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

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

[6] Y. Q. Yuchen Fan, F. K. Soong, and L. He, "Unsupervised speaker adaptation for dnn-based tts synthesis," in *ICASSP*. IEEE, 2016, pp. 5135–5139.

[7] Z. Wu, P. Swietojanski, C. Veaux, S. Renals, and S. King, "A study of speaker adaptation for dnn-based speech synthesis," in *Interspeech*, 2015.

[8] B. Potard, P. Motlicek, and D. Imseng, "Preliminary work on speaker adaptation for dnn-based speech synthesis," Idiap, Tech. Rep., 2015.

[9] J. Yamagishi and T. Kobayashi, "Average-voice-based speech synthesis using hsmm-based speaker adaptation and adaptive training," *IEICE Transactions on Information and Systems*, vol. 90, no. 2, pp. 533–543, 2007.

[10] T. Anastasakos, J. McDonough, R. Schwartz, and J. Makhoul, "A compact model for speaker-adaptive training," in *Spoken Language*, vol. 2. IEEE, 1996, pp. 1137–1140.

[11] C. Leggetter and P. C. Woodland, "Speaker adaptation of continuous density hmms using multivariate linear regression," in *ICSLP*, vol. 94. Citeseer, 1994, pp. 451–454.

[12] O. Abdel-Hamid and H. Jiang, "Fast speaker adaptation of hybrid nn/hmm model for speech recognition based on discriminative learning of speaker code," in *ICASSP*. IEEE, 2013, pp. 7942–7946.

[13] J. T. Zhiying Huang, S. Xue, and L.-R. Dai, "Speaker adaptation of rnn-blstm for speech recognition based on speaker code," in *ICASSP*. IEEE, 2016, pp. 7942–7946.

[14] S. Xue, O. Abdel-Hamid, H. Jiang, and L. Dai, "Direct adaptation of hybrid dnn/hmm model for fast speaker adaptation in lvcsr based on speaker code," in *ICASSP*. IEEE, 2014, pp. 6339–6343.

[15] S. Xue, H. Jiang, L. Dai, and Q. Liu, "Unsupervised speaker adaptation of deep neural network based on the combination of speaker codes and singular value decomposition for speech recognition," in *ICASSP*. IEEE, 2015, pp. 4555–4559.

[16] S.-i. Horikawa, T. Furuhashi, and Y. Uchikawa, "On fuzzy modeling using fuzzy neural networks with the back-propagation algorithm," *IEEE transactions on Neural Networks*, vol. 3, no. 5, pp. 801–806, 1992.

[17] A. Graves, N. Jaitly, and A.-R. Mohamed, "Hybrid speech recognition with deep bidirectional lstm," in *Proceedings of ASRU*, 2013, pp. 273–278.

[18] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788–798, 2011.

[19] J. Kominek and A. W. Black, "The cmu arctic speech databases," in *Fifth ISCA Workshop on Speech Synthesis*, 2004.

[20] K. Oura, "List of modifications made in hts (for version 2.2 beta)," 2011.

[21] H. Kawahara, I. Masuda-Katsuse, and A. De Cheveigne, "Restructuring speech representations using a pitch-adaptive time–frequency smoothing and an instantaneous-frequency-based f0 extraction: Possible role of a repetitive structure in sounds," *Speech communication*, vol. 27, no. 3, pp. 187–207, 1999.

[22] F. Weninger, "Introducing currennt: The munich open-source cuda recurrent neural network toolkit," *Journal of Machine Learning Research*, vol. 16, pp. 547–551, 2015.

[23] N. Brümmer, "The spescom data voice nist sre 2004 system," in *NIST SRE 2004 Workshop, Toledo, Spain*, 2004.

[24] M. A. Przybocki, A. F. Martin, and A. N. Le, "Nist speaker recognition evaluation chronicles-part 2," in *Speaker and Language Recognition Workshop*. IEEE, 2006, pp. 1–6.

[25] A. Strasheim and N. Brümmer, "Sunsdv system description: Nist sre 2008," in *NIST Speaker Recognition Evaluation Workshop Booklet*, 2008.

[26] D. Graff, K. Walker, and A. Canavan, "Switchboard-2 phase ii," *LDC 99S79–http://www. ldc. upenn. edu/Catalog*, 1999.

[27] D. Graff, K. Walker, and D. Miller, "Switchboard cellular part 2," *LDC 2004S07–https://catalog.ldc.upenn.edu/LDC2004S07*, 2004.

[28] R. F. Kubichek, "Mel-cepstral distance measure for objective speech quality assessment," in *Communications, Computers and Signal Processing*, vol. 1. IEEE, 1993, pp. 125–128.