10th ISCA Speech Synthesis Workshop
20-22 September 2019, Vienna, Austria

Speaker Adaptation of Acoustic Model using a Few Utterances in DNN-based Speech Synthesis Systems

Ivan Himawan, Sandesh Aryal*, Iris Ouyang, Shukhan Ng, Pierre Lanchantin*

ObEN, Inc., Pasadena, California, USA

{ivan, sandesh, iris, shukhan, pierre}@oben.com

Abstract

Synthesizing a person's voice from only a few utterances is a highly desirable feature for personalized text-to-speech systems. This can be achieved by adapting an existing speaker-independent model to a target speaker such that the speaker variabilities due to a mismatch between training and testing conditions are minimized. In deep neural network (DNN) based speech synthesis, directly fine-tuning a large number of parameters is susceptible to over-fitting problem, especially when the adaptation set is small. In this paper, we present a novel technique to estimate a speaker-specific model using a partial copy of the speaker-independent model by creating a separate parallel branch stemmed from the intermediate hidden layer of the base network. This allows the fine-tuning of a speaker-specific model to take into account the difference between the target speaker and a speaker-independent model output. Experimental results show that the proposed adaptation method achieves improved audio quality and higher speaker similarity compared to another DNN speaker adaptation technique.

Index Terms: PBFT, SPSS, DNN, speaker adaptation, text-to-speech

1. Introduction

The success of statistical parametric speech synthesis (SPSS) systems [1] is largely driven by the availability of high-quality recordings for training the acoustic model. In recent years, significant quality improvements of text-to-speech (TTS) systems have been due to the progress in the fields of machine learning which has led to development of deep neural network (DNN) based speech synthesis [2, 3, 4]. Typically, the SPSS system's pipeline consists of a text analyzer that converts input text to linguistic features, and a back-end speech synthesizer that maps linguistic features into acoustic features. For the waveform generation part, a vocoder (i.e., WORLD [5], PML [6]) is employed to synthesize speech from the acoustic features. Recently, end-to-end TTS frameworks have become the state of the art for speech synthesis tasks, capable of producing very high-quality voice samples. In the end-to-end frameworks, several specialized modules in the SPSS pipeline are combined together and replaced by a single neural network. Neural TTS systems such as WaveNet [7] directly convert the linguistic features into a waveform. Other systems such as Char2Wav [8] and Tacotron [9] directly map the input text into acoustic features. In Baidu's DeepVoice [10, 11], the entire TTS pipeline is implemented using a similar structure as the traditional TTS system by replacing all modules with neural networks.

A large volume of high-quality recordings is usually required for training acoustic models using neural networks. Often it is difficult to obtain high-quality recordings because of challenging acoustic environments or infrequent speaking styles and sounds to build a new voice. In particular, we are interested to produce TTS models that can be adapted rapidly to an unseen target speaker using a small amount of data at deployment time. Such systems are expected to produce an arbitrary speaker's voice without sacrificing the naturalness of speech and similarity to the target speaker. This is a challenging task, and DNN-based speaker adaptation techniques have been developed to deal with this problem [4, 12, 13, 14, 15]. The recent extension to the end-to-end techniques (i.e., neural voice cloning) have also enabled a new person's voice to be synthesized using a small amount of speech data. Most of these recent approaches employ speaker embedding networks trained to learn characteristics from many different speakers in order to fit a new speaker in the embedding subspace, using back-propagation [16, 17, 18, 19]. However, a couple minutes of adaptation data instead of seconds are required to obtain acceptable results. For example, [16] used Deep Voice 3 [11] architecture and fine-tuned a trained multi-speaker model and the speaker encoder network for whole model adaptation, or speaker encoder network for embedding-only adaptation. Although this approach produced high-quality voice samples, the joint optimization of multi-speaker model and speaker encoder from scratch may not yield good speaker similarity since the speaker encoder tries to minimize the overall generative loss and predicts an average voice instead of a new speaker.

A popular approach for model adaptation of DNN acoustic models is a transformation-based method. This method was originally employed for speaker adaptation in automatic speech recognition (ASR) tasks by augmenting the existing neural networks with new layers [20]. Typically, a new layer is trained with the speaker-specific data, while keeping the rest of network parameters fixed. A similar strategy is employed by learning hidden unit contributions (LHUC) by introducing speaker-specific parameters to transform a speaker independent feature space to a speaker-dependent feature space using a small amount of adaptation data [21]. In LHUC, the learned feature detectors from the speaker-independent model are reused while adjusting the contributions of the hidden units in the model using the adaptation data. As Parker et. al. noted in their paper [22], if the speaker-independent (SI) model does not contain feature detectors that are required by the new speaker, the performance of LHUC will degrade significantly. This usually happens when the new speaker is different from the speakers used to train the SI model.

Continued fine-tuning of a SI model using adaptation utterances is also a viable alternative. However, the model with more than million parameters can lead to over-fitting problem when fine-tuned using a small amount of data. In addition, due to the small amount of adaptation data, known regularization techniques are susceptible for over-smoothing the spectrogram, resulting in low quality speech. In this work, we propose a spe-

*Equal Contribution.

10.21437/SSW.2019-9

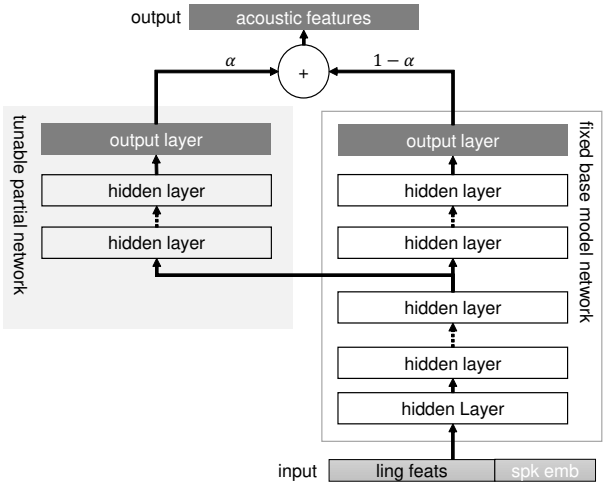

Figure 1: *Illustration of the parallel branched network structure for speaker adaptation of acoustic model. The network input features are linguistic features (ling feats) and speaker embedding (spk emb).*

cial modification in the network structure along with the training strategy to address those limitations of the continued fine-tuning approach. Our proposed solution involves a weighted sum of the outputs from a pretrained SI model and a partial copy of it as shown in Figure 1. In this configuration, we refer the SI model network as the base model network and the partial copy as the parallel branch. During fine-tuning, we only allow updating parameters in the parallel branch while keeping the network in the base model network unchanged, hence the name parallel branch fine tuning (PBFT). The idea is to adjust the parallel branch to account for a scaled version of the differences between the target acoustic features and the base model estimates. We hypothesize that a highly regularized training process that learns a smooth function might be less detrimental to the quality of speech when the function is used to estimate the differences instead of the acoustic features themselves.

The contributions of this paper are two-folds. First, we introduce a PBFT method to overcome the shortcomings of LHUC and traditional model adaptation approaches. Second, we investigate the generalization capability of the network using a few seconds of adaptation data (i.e., 5, 10, 20, and 35 utterances). Subjective listening tests are also conducted to assess the quality and speaker similarity of generated speech. This paper is organized as follows: Section 2 describes LHUC and PBFT methods. Experimental setup is described in Section 3. Results are presented and discussed in Section 4. Finally, the study is concluded in Section 5.

2. Related works

Adaptive TTS systems can be built by training the multi-speaker synthesis model using multiple known speakers and then adapting the speaker-independent model to new speakers [23]. Speaker adaptation techniques for DNN-based TTS systems can be broadly categorized into three approaches. The first approach is model adaptation by fine-tuning the neural network parameters directly to model the unseen speaker's data better. The second one is to augment input features with the speaker-specific information, which can be fed to one or many layers of DNN and jointly optimized with the rest of DNN parameters during DNN training. This speaker-specific information is typically represented as a speaker code such as one-hot vector [24, 25], speaker identity vector (i-vector) [4, 26], or the embedding vector [17]. The third approach is to perform feature space transformation to transform the predicted vocoder parameters to the target speaker's parameters [4, 27]. All these approaches may be combined to improve the adaptation performance when adapting the SI model [4].

In DNN-based speech synthesis, LHUC has been employed to perform speaker adaptation, and recently, it is used to adapt expressive speech from one single speaker to a new speaker using only neutral speech [22]. LHUC adaptation makes it attractive for adapting acoustic model to a new speaker using only a few utterances. It avoids over-fitting. However, it has been pointed out that the LHUC approach does not change any specific feature of each hidden node in the speaker-independent model is specialized on [21]. It only changes the weight of each feature and that may be a limitation when the target speaker is different from speakers used to train the SI model [21]. Another adaptation technique, developed by Parker et al. [22] and called Hidden layer Augmentation (HLA), augments hidden layers with expression and speaker specific nodes. These augmented nodes are fully connected to the preceding and succeeding layers allowing them to learn new features. Results of their study found such adaptation are more effective in speaker and expression adaptation than LHUC method. Moreover, this study was conducted with hundreds of seconds of adaptation utterances instead of a few seconds using significantly higher number of parameters.

3. Methods

3.1. Adaptation using LHUC

In the standard feed-forward neural network, the output of the l-th hidden layer, $\mathbf{h}_l$ can be defined by the following equation,

$$\mathbf{h}_l = \sigma(\mathbf{W}_l \mathbf{h}_{l-1} + \mathbf{b}_l) \tag{1}$$

where $\mathbf{W}_l$ denotes the matrix of connection weights from the nodes in $l-1$-th to l-th layers, and $\mathbf{b}_l$ is the additive bias vector at the l-th layer. σ is the non-linear activation function (such as *tanh* or *sigmoid*). The LHUC adaptation aims to adjust the SI model parameters so that they generalize better to unseen speakers using small amount of adaptation data [21]. To achieve this, LHUC introduces speaker dependent parameters to the existing network parameters where the learned feature detectors from SI network are frozen during training. Specifically, LHUC introduces speaker-specific parameters $\theta^k = \{\mathbf{r}_l^k, \ldots, \mathbf{r}_L^k\}$, re-parameterized by a function $f(.)$, into SI network with the set of parameters $\Theta = \{\mathbf{W}_1, \ldots, \mathbf{W}_L, \mathbf{b}_1, \ldots, \mathbf{b}_L\}$. The $\mathbf{r}_l^k$ denotes the vector of the speaker-specific parameters for the l^{th} hidden layer for speaker k. Hence, the modified hidden layer output of the l-th layer, $\mathbf{h}_l$ is obtained as:

$$\mathbf{h}_l^k = f(\mathbf{r}_l^k) \circ \sigma(\mathbf{W}_l \mathbf{h}_{l-l}^k + \mathbf{b}_l), \tag{2}$$

where $\circ$ is an element-wise multiplication. During adaptation, θ^k is optimized using the standard error back-propagation procedure while keeping the Θ fixed for speaker k.

3.2. Adaptation using PBFT

The parallel branch fine-tuning (PBFT) adaptation method extends the structure of the base model network by adding a parallel branch that is tuned for speaker specific characteristics as shown in Figure 1. The parallel branch is a partial copy of the

base model feed-forward network stemmed from a predefined intermediate hidden layer $\mathbf{h}_l$. The input to this partial parallel branch is the activation at the $\mathbf{h}_{l-1}$ layer of the base model network. Given a weight parameter α, such that $0 < \alpha < 1$, the output of the network is given by the weighted sum,

$$\hat{\mathbf{y}} = \alpha \hat{\mathbf{y}_{\mathbf{pb}}} + (1 - \alpha)\hat{\mathbf{y}_{\mathbf{si}}}, \quad (3)$$

where $\hat{\mathbf{y}_{\mathbf{pb}}}$ and $\hat{\mathbf{y}_{\mathbf{si}}}$ are the outputs from the tune-able parallel branch and the frozen base model, respectively. The weight parameter α and the number of layers to include in the partial parallel network are the design choices guided by the amount and the quality of the adaptation utterances from the target speaker. During adaptation phase, the base model parameters are fixed and only the parameters in the parallel branch are updated.

4. Experiments

4.1. Database and features

In this work, we used a combined VCTK and Appen ASR corpus to train the base model of male speakers. The total duration is about 59.5 hours, and the number of utterance is 77k pooled from 100 unique speakers. The acoustic features were extracted from 48 kHz waveform at 5 millisecond interval using WORLD [5]. These features consisted of 60 Mel-Cepstral coefficients (MCCs), 3 band aperiodicities (BAPs), and fundamental frequency F_0 on log scale. Thus, the output features for the neural networks consisted of MCCs, BAPs, and log F_0 with their delta and delta-deltas features, and an additional voiced/unvoiced binary feature. The input linguistic features consisted of 651-dim features. 648 of these represented linguistically-related context including quinphone identity, parts-of-speech, and positional information of phoneme, syllable and word. The other 3 are within-phone positional information [28].

We also use speaker embeddings as an additional input to encode speaker identities. In our case, the speaker identity of the new target speaker is represented by fixed-dimensional speaker embedding from a speaker encoder network instead of i-vectors in the i-vector based speech synthesis system. We concatenated 200-dim embeddings vectors to the linguistic features to form a total of 851-dim input feature vectors for all neural networks [17].

Our speaker adaptation set consists of two male speakers, *SPK1 and SPK2*. Each speaker recorded a total number of 35 utterances using their cellphones. To compare the effect of the available adaptation data on the performance of adaptation techniques, we created four sets of adaptation utterances, namely, *Utt05, Utt10, Utt20, and Utt35*, such that *Utt05* $\subset$ *Utt10* $\subset$ *Utt20* $\subset$ *Utt35*. Table 1 shows the number of adaptation utterances and the total duration for each target speaker. For each adaptation utterance group, 80% of the utterances were used for training and the remaining 20% for validation. All the objective scores presented in this work are based on the five validation utterances which were not used for training in any of the group.

4.2. DNN configuration

The acoustic models for training the multi-speaker model were feed-forward DNN with ten hidden layers. The number of neurons for each hidden layer were 1024, 512, 512, 256, 256, 512, 512, 512, 1024, 1024, respectively. A hyperbolic tangent was used as the activation function for each hidden layer, followed by a linear activation at the output layer. The batch-normalization is applied to each hidden layer except the first input layer. We initialized the new weight parameters randomly and trained the models to minimize mean square error using stochastic gradient descent, and a batch size of 1024. Learning rate was fixed at 0.001, warm-up momentum was 0.4, drop-out rate was 0.02, and the number of training epochs was 100. For adaptation experiments, LHUC weights were applied to all hidden units and the learning rate was fixed at 0.1. We copied the last four hidden layers from the base network to create a parallel branch for PBFT, the PBFT weight α was set to 0.8 and the learning rate was fixed at 0.001. The decision process involved in choosing the specific PBFT weight parameter is discussed in detail in the next subsection. Different learning rates were used for LHUC and PBFT based on the cross validation errors observed in our preliminary tests. The difference in optimal learning rates for these two methods may be related to the difference in the number of model parameters (about 6000 in LHUC and 2.3M in PBFT). We used Merlin toolkit for training acoustic models [28].

Table 1: *The number of adaptation utterances and the total duration (in seconds) in the speaker adaptation datasets. The speech-only duration (the number inside the bracket) is obtained after applying voice activity detection (VAD).*

Adapt. set	Num. utt.	*SPK1* dur.	*SPK2* dur.
Utt05	5	17.6 (15.3)	19.8 (17.5)
Utt10	10	47.3 (42.3)	50.8 (45.1)
Utt20	20	101.8 (92.0)	107.1 (94.4)
Utt35	35	185.0 (169.2)	194.7 (168.7)

4.3. Selection of branch weight parameter

Statistical smoothing in parametric speech synthesis is a well known issue that causes muffling in speech synthesis [2]. One of the symptoms of such smoothing is the lower global variance in the estimated acoustic features compared to that of natural speech [27]. We introduced the branch weight parameter (α) in PBFT model as a mechanism to control such smoothing. To investigate this expected effect of (α) in the model, we looked at the global variance of MCCs and the Mel Cepstral Distortion (MCD) for the models adapted using $\alpha = \{0.1, 0.4, 0.8, 0.99\}$. Ideally, the estimated parameters from the model is expected to match the global variance of the natural utterances while minimizing the MCD.

We randomly chose SPK2 and corpus size *Utt10* for this ex-

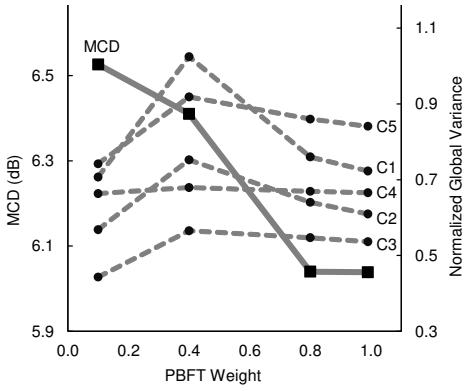

Figure 2: *Effect of the branch weight parameter α in model accuracy and global variance.*

Table 2: *Objective results of LHUC and PBFT adaptation techniques for different adaptation corpus sizes.*

	SPK1				*SPK2*			
	LHUC		PBFT		LHUC		PBFT	
Adapt. set.	MCD (dB)	F_0 RMSE (Hz)	MCD (dB)	F_0 RMSE (Hz)	MCD (dB)	F_0 RMSE (Hz)	MCD (dB)	F_0 RMSE (Hz)
Utt05	8.07	52.36	8.09	49.40	6.57	24.10	6.56	24.56
Utt10	8.02	50.85	7.54	47.93	6.57	24.64	6.04	22.27
Utt20	8.05	47.32	7.51	45.33	6.52	25.50	6.02	21.89
Utt35	7.70	49.84	7.35	44.60	6.41	25.77	5.78	21.35

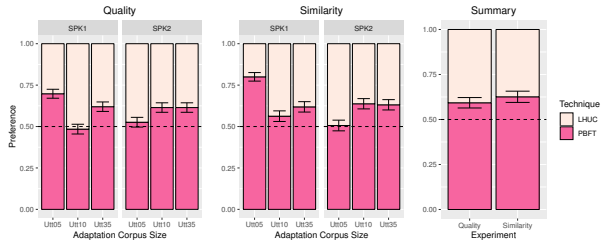

Figure 3: *Results of subjective evaluation on quality and similarity of LHUC vs. PBFT. The error bars represent 1 standard error.*

periment. From all the adapted models, acoustic features were estimated for the same set of test utterances. Figure 2 shows a plot of MCDs and the normalized global variance for few MCCs ($C1 - C5$) against the PBFT weight parameters. Normalized GVs were computed by dividing the average GVs for each test utterances by the corresponding average GVs for the natural recordings of the same utterance. Normalized GVs less than 1.0 represents the degree of statistical smoothing occurred on the estimated MCCs.

The figure shows that MCD decreases as α increases, validating the fact that the more weight we give to the parallel branch, the model accuracy keeps increasing. On the other hand, the normalized GV plots comes closest to being 1.0 at $\alpha = 0.4$ and then drops as α increases for all MCCs except $C4$ [1]. When α is close to 1.0, higher weights result in lower spectral distortion but smoother MCCs. For this particular case, we noted that the MCD is almost the same for $\alpha = 0.8$ and $\alpha = 0.99$. Given the similar MCD, we decided to select $\alpha = 0.8$ instead of $\alpha = 0.99$ because of the higher global variance in the former. We kept the same weight parameter in all our subsequent adaptation models in this study, regardless of the speaker and adaptation corpus sizes.

5. Results and discussion

5.1. Objective evaluation

Following [4], we computed MCD and root mean square error (RMSE) of fundamental frequency (F_0) for each test utterances as the objective measures of model performance. Table 2 shows objective measures for LHUC and PBFT in order to analyze the performance of individual adaptation techniques. Overall, PBFT obtained lower MCDs and F_0 RMSE distortions compared to LHUC for both speakers. For adaptation utterance group *Utt05*, PBFT and LHUC gave comparable performances. However, PBFT achieved the lowest distortions when the number of adaptation utterances increases. For LHUC, we found that the distortions are reduced slightly when we increased the number of adaptation utterances from 5 to 10, or 10 to 20. On the other hand, reduction of distortions is noticeable for PBFT when we increased the number of adaptation utterances from 5 to 10, and 20 to 35. For both PBFT and LHUC, small performance differences are observed between 10 and 20 utterances.

5.2. Subjective evaluation

The objective scores allow us to compare the accuracy of models but the model accuracy does not necessarily correlate with the perceived quality of the synthesized speech. We performed four subjective listening experiments to evaluate the model in terms of the quality and the voice similarity.

For subjective listening tests, phonetically balanced sentences[2] that are not part of the training utterances were generated from all 12 trained acoustic models (2 speakers $\times$ 2 adaptation techniques $\times$ 3 adaptation utterance sets). Out of the four adaptation utterance sets, we excluded *Utt20* from the subjective tests because its objective performance was comparable to that of *Utt10*. Same phoneme durations, estimated by an internal system, were shared by all 12 models so that the listeners could focus only on the performance of the acoustic model. The internal system used for duration estimation in this experiment is a gradient boosting regression tree model trained on specifically designed single speaker corpus. Speech samples are available online [3]. We evaluated these models through a series of perceptual listening experiments on Amazon Mechanical Turk, a crowd sourcing tool.

Participants were native speakers of English residing in the United States. In all experiments, participants were asked to compare sound clips generated by two different models and select one according to the task instructions. An additional five pairs of audios were included as decoys in each survey to control the quality of the submissions. In the decoys, one of the two sound clips in each pair was replaced by copy-synthesis of the same sentence's natural recording from the same speaker. We discarded submissions with less than 80% accuracy for decoys. All test pairs and decoys were randomized for each participant. The order of the two sound clips in a pair was also randomized.

5.2.1. LHUC vs. PBFT quality evaluation

In the first listening experiment, we compared the performance of the two adaptation techniques in terms of the acoustic quality.

[1] Low GV at $\alpha = 0.1$ might be due to the difference in GV of the target speaker and the estimates from the SI base model which dominates the output.

[2] https://www.cs.columbia.edu/ hgs/audio/harvard.html

[3] https://oben-ssw10.github.io/demo/

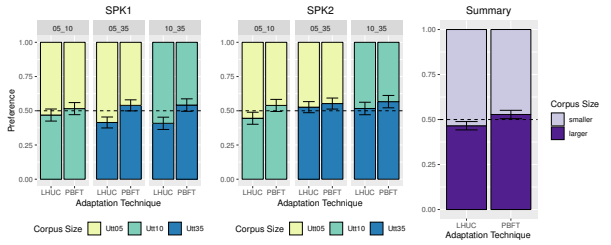

Figure 4: *Results of subjective evaluation on the quality of TTS built from different adaptation corpus sizes.*

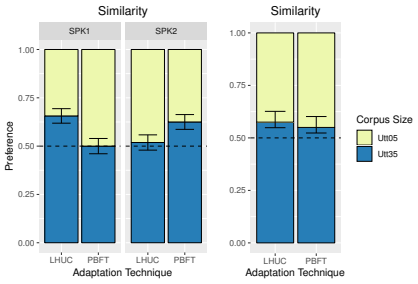

Figure 5: *Results of subjective evaluation on the similarity of TTS built from different adaptation corpus sizes.*

The two sound clips in a pair were of the same sentence generated by the PBFT and LHUC adaptation models, respectively. Participants were asked to choose the one that sounded better to them. Eight test sentences were used, counterbalanced in a 2 (speakers) x 3 (adaptation utterance sets) design across two versions of the survey, hence a total of 48 test audio pairs. To avoid fatigue, each participant was given one of the two versions. 48 submissions were included in the data analysis; 7 submissions were discarded due to their low accuracy in the decoys.

Figure 3 (left) shows the subjective tests results of PBFT and LHUC using different adaptation corpus sizes. In most of the cases, PBFT is preferred over LHUC in terms of quality. This trend holds across speakers and adaptation corpus sizes ($z = 4.763, p < 0.001$).

5.2.2. *LHUC vs. PBFT speaker similarity evaluation*

In the second listening experiment, we compared the two adaptation techniques in terms of similarity between the synthesized speech and the target speaker in an ABX format. A reference utterance was presented along with the two utterances generated from PBFT and LHUC. Participants were asked to select the one in the pair that was spoken by the same speaker as the reference utterance. The reference was copy-synthesis of a random original adaptation utterance from the same speaker, linguistically different from the two utterances being compared. We artificially degraded the quality of the reference by adding noise so that the listener would not pick the sound clip with better acoustic quality to match the reference and ignore the voice similarity. 40 submissions were included in the data analysis; 3 submissions were discarded due to their low accuracy in the decoys.

Similar to the quality evaluation results, the summary plot in Figure 3 shows that PBFT is again preferred over LHUC in similarity evaluation. This trend holds across speakers and adaptation corpus sizes ($z = 6.550, p < 0.001$), indicating that PBFT is a better adaptation technique than LHUC in matching the voice color of the target speaker.

5.2.3. *Effect of training corpus size on quality*

In the third listening experiment, we evaluated how the adaptation techniques performed in terms of synthesis quality when trained on different amount of adaptation data. The experiment consisted of three surveys, differing in the size of the adaptation corpus: *Utt05* compared with *Utt10*, *Utt10* compared with *Utt35*, and *Utt05* compared with *Utt35*, respectively. Similar to the LHUC vs. PBFT quality evaluation, the two sound clips in a pair were generated using the same sentence, and participants were asked to choose the one that sounded better to them. Eight test sentences were used, counterbalanced in a 2 (speakers) x 2 (adaptation techniques) design, hence a total of 32 test audio pairs in every survey. Each participant was given one of the three surveys. 50 submissions were included in the data analysis; 8 submissions were discarded due to their low accuracy in the decoys.

Figure 4 shows participants' choices between the smaller and the larger adaptation corpus sizes in each survey. Overall, we observe a stronger preference for larger corpus sizes in PBFT compared to LHUC ($z = 3.150, p < 0.01$). When we analyze the trends in PBFT and LHUC separately, we find that larger corpus sizes are preferred for PBFT ($z = 2.401, p < 0.05$), while smaller corpus sizes are preferred for LHUC ($z = 2.049, p < 0.05$). It shows that PBFT can take advantage of more adaptation data to improve quality but it is not the case for LHUC. In fact, in SPK1, the quality is degraded for LHUC using more adaptation data. This may be caused by a large variation of speaking style that we observed in a small pool of adaptation utterances recorded by SPK1. Also, study in [4] reported small quality improvement in terms of multiple stimuli with hidden reference and anchor (MUSHRA) scores when utterance adaptation data increases 10 times (i.e., 10 vs. 100).

5.2.4. *Effect of training corpus size on speaker similarity*

In the fourth listening experiment, we seek to determine how the adaptation techniques perform in matching the voice of the target speaker with increased amount of adaptation data. Based on the objective scores and the results from the subjective quality evaluation, we decided to assess the effect of adaptation corpus size in the extreme cases only (*Utt05* vs. *Utt35*). The survey design was very similar to the other experiments described above. Eight test sentences counterbalanced in speakers and adaptation techniques, creating a total of 32 test audio pairs. 20 submissions were included in the data analysis; no submission failed the decoys.

The plots in Figure 5 shows that *Utt35* is preferred over *Utt05* across speakers and adaptation techniques ($z = 3.131, p < 0.01$). This trend also holds for each adaptation techniques separately (LHUC: $z = 3.114, p < 0.01$; PBFT: $z = 2.230, p < 0.05$). These results suggest that more adaptation data helps improving speaker similarity for both LHUC and PBFT.

6. Conclusions

In this paper, we proposed a technique, referred to as parallel branch fine tuning (PBFT), for speaker adaptation of DNN-based acoustic model in a TTS systems when only a few short utterances are available from a new speaker. The proposed method involves fine-tuning of a partial copy of the speaker-independent acoustic model network branched off from an intermediate hidden layer where outputs from each branches are multiplied with predetermined weights and added. In compar-

ison with LHUC (a method specifically designed for adapting acoustic model with a small amount of data), PBFT is found better in both acoustic quality and the speaker similarity regardless of the adaptation utterance sets.

We see a tendency that, with the use of more adaptation utterances (for a small corpus size up to 35 utterances), the quality of PBFT improves. We observed a similar trend for speaker similarity too. In contrast, within the range of our adaptation corpus sizes, we did not notice a consistent quality improvement with the increase in adaptation data in LHUC, even though it improved the speaker similarity. Unlike LHUC, PBFT adjusts a multilayer nonlinear network with significantly large number of parameters during adaptation. Such a large adaptation network allows the model to learn complex contextual relationships, making PBFT more appropriate for adapting to a very different new speaker than LHUC. In addition, the large number of parameters also allows the model to improve with the increase in adaptation data.

Although we observed some important advantages of the proposed method in this work, a large scale study is necessary to validate if these findings generalize over a larger varieties of target speakers including female speakers, children, and elderly people. Further studies are also needed to understand how the branch weight parameters of the model affect the performance for different speakers and adaptation corpus sizes. We also plan to investigate if the branch weights can be applied as a possible trade-off between speaker similarity and the acoustic quality.

7. Acknowledgements

We would like to thank our colleagues Carol Figueroa and Sirui Xu for their contributions in the development of some of the tools used in this research projects. Special thanks also go to Sam Kang, Kyung-Min Kim, Sang-Ho Lee, and Ethan Sherr-Ziarko for their invaluable help in survey administration.

8. References

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

[2] H. Zen, A. Senior, and M. Schuster, "Statistical parametric speech synthesis using deep neural networks," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, 2013, pp. 7962–7966.

[3] Y. Fan et al., "Multi-speaker modeling and speaker adaptation for DNN-based TTS synthesis," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, April 2015, pp. 4475–4479.

[4] Z. Wu, et. al, "A study of speaker adaptation for DNN-based speech synthesis," in *Proceedings of Interspeech*, 2015, pp. 879–883.

[5] M. Morise, F. Yokomori, and K. Ozawa, "WORLD: a vocoder-based high-quality speech synthesis system for real-time applications," *IEICE Transactions on Information and Systems*, vol. 99, no. 7, pp. 1877–1884, 2016.

[6] G. Degottex, P. Lanchantin, and M. Gales, "A log domain pulse model for parametric speech synthesis," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 1, pp. 57–70, Jan 2018.

[7] A. van den Oord, et al., "Wavenet: A generative model for raw audio," *arXiv preprint arXiv:1609.03499*, 2016.

[8] J. Sotelo, et al., "Char2Wav: End-to-end speech synthesis," in *Proceedings of International Conference on Learning Representations (ICLR)*, 2017.

[9] Y. Wang et al., "Tacotron: Towards end-to-end speech synthesis," in *Proceedings of Interspeech*, 2017, pp. 4006–4010.

[10] S. Ö. Arik et al., "Deep voice: Real-time neural text-to-speech," in *Proceedings of the 34th International Conference on Machine Learning*, 2017, pp. 195–204.

[11] W. Ping et al., "Deep voice 3: Scaling text-to-speech with convolutional sequence learning," in *Proceedings of International Conference on Learning Representations (ICLR)*, 2018.

[12] B. Li and H. Zen, "Multi-language multi-speaker acoustic modeling for LSTM-RNN based statistical parametric speech synthesis." in *Proceedings of Interspeech*, 2016, pp. 2468–2472.

[13] S. Pascual and A. Bonafonte, "Multi-output RNN-LSTM for multiple speaker speech synthesis with α-interpolation model." in *Proceedings of the 9th ISCA Speech Synthesis Workshop*, 2016, pp. 112–117.

[14] H. T. Luong, et al., "Adapting and controlling DNN-based speech synthesis using input codes," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2017, pp. 4905–4909.

[15] R. Doddipatla, N. Braunschweiler, and R. Maia, "Speaker adaptation in DNN-based speech synthesis using d-vectors." in *Proceedings of Interspeech*, 2017, pp. 3404–3408.

[16] S. Arik et al., "Neural voice cloning with a few samples," in *Advances in Neural Information Processing Systems*, 2018, pp. 10 019–10 029.

[17] Y. Jia, et al., "Transfer learning from speaker verification to multispeaker text-to-speech synthesis," in *Advances in Neural Information Processing Systems*, 2018, pp. 4480–4490.

[18] E. Nachmani, et al., "Fitting new speakers based on a short untranscribed sample," in *Proceedings of International Conference on Machine Learning*, 2018, pp. 3680–3688.

[19] Y. Chen et al., "Sample efficient adaptive text-to-speech," in *Proceedings of International Conference on Learning Representations (ICLR)*, 2019.

[20] B. Li and K. C. Sim, "Comparison of discriminative input and output transformations for speaker adaptation in the hybrid NN/HMM systems," in *Proceedings of Interspeech*, 2010.

[21] P. Swietojanski and S. Renals, "Learning hidden unit contributions for unsupervised speaker adaptation of neural network acoustic models," in *IEEE Spoken Language Technology Workshop (SLT)*, Dec 2014, pp. 171–176.

[22] J. Parker, Y. Stylianou, and R. Cipolla, "Adaptation of an expressive single speaker deep neural network speech synthesis system," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, April 2018, pp. 5309–5313.

[23] J. Yamagishi, et al., "Analysis of speaker adaptation algorithms for HMM-based speech synthesis and a constrained SMAPLR adaptation algorithm," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 17, no. 1, pp. 66–83, Jan 2009.

[24] H. Luong et al., "Adapting and controlling DNN-based speech synthesis using input codes," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, March 2017, pp. 4905–4909.

[25] N. Hojo, Y. Ijima, and H. Mizuno, "DNN-based speech synthesis using speaker codes," *IEICE Transactions on Information and Systems*, vol. 101, no. 2, pp. 462–472, 2018.

[26] M. Wan, G. Degottex, and M. J. F. Gales, "Integrated speaker-adaptive speech synthesis," in *IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, Dec 2017, pp. 705–711.

[27] T. Toda, A. W. Black, and K. Tokuda, "Voice conversion based on maximum-likelihood estimation of spectral parameter trajectory," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 8, pp. 2222–2235, 2007.

[28] Z. Wu, O. Watts, and S. King, "Merlin: An open source neural network speech synthesis system." in *Proceedings of the 9th ISCA Speech Synthesis Workshop*, 2016, pp. 202–207.