# Multi-Accent Adaptation based on Gate Mechanism

*Han Zhu[12], Li Wang[1], Pengyuan Zhang[12], Yonghong Yan[123]*

[1]Key Laboratory of Speech Acoustics and Content Understanding, Institute of Acoustics, China
[2]University of Chinese Academy of Sciences, China
[3]Xinjiang Laboratory of Minority Speech and Language Information Processing, Xinjiang Technical
Institute of Physics and Chemistry, Chinese Academy of Sciences, China

{zhuhan,wangli,zhangpengyuan,yanyonghong}@hccl.ioa.ac.cn

## Abstract

When only a limited amount of accented speech data is available, to promote multi-accent speech recognition performance, the conventional approach is accent-specific adaptation, which adapts the baseline model to multiple target accents independently. To simplify the adaptation procedure, we explore adapting the baseline model to multiple target accents simultaneously with multi-accent mixed data. Thus, we propose using accent-specific top layer with gate mechanism (AST-G) to realize multi-accent adaptation. Compared with the baseline model and accent-specific adaptation, AST-G achieves 9.8% and 1.9% average relative WER reduction respectively. However, in real-world applications, we can't obtain the accent category label for inference in advance. Therefore, we apply using an accent classifier to predict the accent label. To jointly train the acoustic model and the accent classifier, we propose the multi-task learning with gate mechanism (MTL-G). As the accent label prediction could be inaccurate, it performs worse than the accent-specific adaptation. Yet, in comparison with the baseline model, MTL-G achieves 5.1% average relative WER reduction.

**Index Terms**: accented speech recognition, multi-accent, model adaptation, multi-task learning, gate mechanism

## 1. Introduction

Accent is known to be one of the key factors to degrade the performance of automatic speech recognition (ASR) systems [1]. Despite great progress over the last few years, the performance gap between the standard and the accented speech recognition remains large. Accent is mainly caused by non-native speakers or native speakers who speak the dialects of the language. The latter is the primary source of mandarin accents [2]. In real-world applications, we often deal with multi-accent problem because the ASR system would be used by speakers with a diverse set of accents.

There is normally a large amount of standard mandarin speech data, while accented mandarin speech data is relatively less. In order to utilize all these data well, the conventional approach would be to train a baseline model with a large amount of standard mandarin speech data and then adapt the baseline model to a specific accent with respective data to produce the accent dependent model, which we will refer to as accent-specific adaptation.

In terms of adaptation methods, traditional ones for GMM-HMM model, such as MLLR [3] and fMLLR [4], have shown great performance. As deep neural network (DNN) was shown to be more powerful than GMM in acoustic model for large vocabulary continuous speech recognition systems [5], there were many works conducted on DNN adaptation. Selective fine-tuning [6] is a very intuitive method to alleviate the data

mismatch problem. Linear transformation, such as linear input network (LIN) [7], linear hidden network (LHN) [8] and linear output network (LON) [7] were shown to be effective when adaptation data is relatively less. Regularization based methods [9][10] were introduced to solve the overfitting problem during adaptation. Furthermore, cluster adaptive training (CAT) [11], factorized hidden layer adaptation (FHL) [12], learning hidden unit contributions (LHUC) [13] and speaker code [14] were proposed to translate the original problem into evaluation of a small number of parameters during adaptation.

Accent-specific adaptation is cumbersome considering that multiple accent dependent models will be produced and we have to find the best configuration for them individually. Therefore, an integrated adapted model would be desirable. To that end, we can just adapt the baseline model to multiple target accents simultaneously with multi-accent mixed data in an integrated model, which we will refer to as multi-accent adaptation. Adaptation methods like CAT, FHL, and speaker code require multi-class quantitatively balanced data during the baseline model training procedure, which makes it impossible to transfer these methods to multi-accent adaptation. In terms of other adaptation methods like selective fine-tuning and linear transformation, we found it empirically that multi-accent adaptation based on them can improve speech recognition performance, but still performs worse than the accent-specific adaptation.

In this literature, we proposed two novel approaches to promote the performance of multi-accent adaptation. Our work concentrated on the neural network part of the acoustic model. We kept the rest of the ASR system, such as HMM and language model, to be unchanged. Firstly, we proposed accent-specific top layer with gate mechanism (AST-G). In AST-G, on the one hand accent-specific top layer [15] is used to model the distinctive characteristics of multiple target accents via different parameter sets, on the other hand gate mechanism [16] is used to modulate the network's internal representations in an accent-specific way with the help of accent category labels. Experiments showed that AST-G can achieve 9.8% and 1.9% average relative WER reduction compared with the baseline model and accent-specific adaptation respectively. In order to solve one significant shortcoming of AST-G which is that accent category labels are required for inference, we proposed multi-task learning [17] with gate mechanism (MTL-G), in which the primary task is context-dependent tied phone state classification and the secondary task is accent classification. The output of the secondary task network is used as the accent category label and fed into the gate unit of the primary task network.

The rest of the paper is organized as follows, in Section 2 and Section 3, we describe our proposed approaches in detail. Then, we evaluate our proposed approaches and compare them with commonly used adaptation methods in Section 4, and con-

clude the study in Section 5.

## 2. Accent-specific top layer with gate mechanism (AST-G)

In this approach, we assume a priori that accent category labels are available for both adaptation and inference. As figure 1 demonstrated, gate unit (yellow) and accent-specific top layers (green) are added on the basis of the baseline model (blue) before adaptation. We use bidirectional long short-term memory (BLSTM) recurrent neural network as the specific DNN in our experiments for its outstanding performance in acoustic model.

### 2.1. Gate mechanism

The speech spoken by speakers with different accents differs in the phonetic pronunciation. Utilizing these differences in acoustic model was proven to be useful [18]. However, using manufactured phonetic information is quite subjective and cumbersome. It would be convenient if we only provide accent category labels and let the neural network model the difference in phonetic information between target accents. And our gate mechanism is a way to do so. Our work is similar to the one in [16] which is conducted on the multilingual ASR task. We use accent category labels to modulate the network's internal representations in an accent-specific way. To do so, the output of the $i_{th}$ layer $h_i$ and the accent category label (one-hot vector) $v_a$ are fed into the gate unit. After some operations, the gate value $g(h_i, v_a)$ is passed to the $(i+1)_{th}$ layer. Several concrete realizations of the gate unit can be used. The operations of these different gate units are described below.

- Addition (which we will refer to as the GATE I),

$$g(h_i, v_a) = h_i + V v_a + b \qquad (1)$$

  where $V$ is a matrix of size $M \times N$ if dimensions of $h_i$ and $v_a$ are $M$ and $N$ respectively, and $b$ is a bias vector.

- More complex addition (which we will refer to as the GATE II),

$$g(h_i, v_a) = U h_i + V v_a + b \qquad (2)$$

  where $U$ is an identity matrix.

- Addition with a sigmoid activation function (which we will refer to as the GATE III),

$$g(h_i, v_a) = \sigma(h_i + V v_a + b) \qquad (3)$$

  Where $\sigma()$ denotes the sigmoid function.

- Elementwise multiplication (which we will refer to as the GATE IV),

$$g(h_i, v_a) = h_i \cdot V v_a + b \qquad (4)$$

- Addition and elementwise multiplication (which we will refer to as the GATE V),

$$g(h_i, v_a) = h_i \cdot (h_i + V v_a + b) \qquad (5)$$

### 2.2. Accent-specific top layer

The accent-specific top layer [15] was motivated by the shared hidden layers of deep neural network in the multilingual speech recognition [19]. It can be seen as a similar way to gate mechanism to model the distinctive characteristics of multiple target
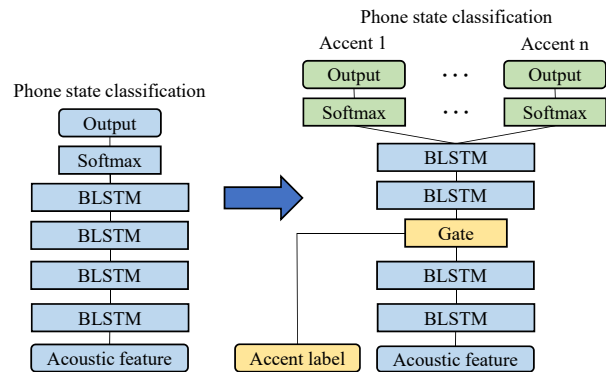


Figure 1: *Accent-specific top layer with gate mechanism (AST-G).*

accents. Before adaptation, the top layer of the baseline model is duplicated to build multiple accent-specific top layers. During adaptation, in our approach, accent-specific top layers are updated together with the extra parts of the network, which is different from the original one which only update the parameters of accent-specific top layers [15]. The accent-specific top layer will only be updated with the specific target accented speech data, while the shared bottom layers will be updated with all target accented speech data. After adaptation, the inference of a specific target accented speech will be done using the respective accent-specific top layer and the shared bottom layers.

The shared accent independent bottom layers allow maximal knowledge transfer between multiple target accents. This is especially important when only small amount of accented speech data is available [15].

### 2.3. AST-G

AST-G uses the above two methods to further improve the performance, which make sense because they have the same goal to model multiple target accents accurately by using either accent category labels or accent-specific parameter sets.

## 3. Multi-task learning with gate mechanism (MTL-G)

This approach is demonstrated in Figure 2. The network jointly predicts context-dependent tied phone states (the primary task) and accent category labels (the secondary task). Unlike AST-G, the extra information fed into the gate unit is the output of the secondary task network, which can be seen as the soft accent category label. Therefore, we only need to know the accent category of the utterances in the adaptation procedure. Before adaptation, the gate unit and the secondary task network (yellow) are added on the basis of the baseline model (blue).

During adaptation, the whole network is trained using back-propagation with the following loss function,

$$L_{total} = (1 - \lambda) L_{primary} + \lambda L_{secondary} \qquad (6)$$

where $\lambda$ is a weight hyperparameter that is used to linearly interpolate the individual loss terms.

After adaptation, the entire network will predict the tied phone states together.

The promising aspect of MTL-G is that the accent category label won't be required for inference, which would be convenient in real-world applications.
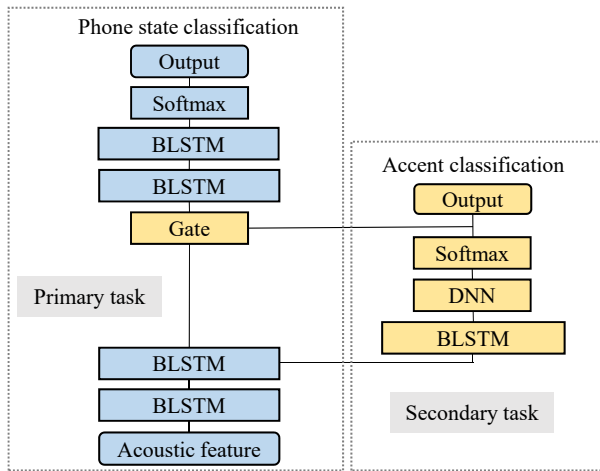
Figure 2: *Multi-task learning with gate mechanism (MTL-G).*

Table 1: *WERs (%) of models using selective fine-tuning and linear transformation*

| model | AH | BJ | SX | YN | AVE |
|---|---|---|---|---|---|
| baseline | 28.9 | 9.8 | 30.9 | 24.3 | 23.5 |
| fine-tune(1) | 28.4 | 10.2 | 30.3 | 24.1 | 23.3 |
| fine-tune(2) | 28.0 | 10.0 | 29.9 | 23.9 | 23.0 |
| fine-tune(3) | 27.6 | 9.9 | 29.6 | 23.7 | 22.7 |
| fine-tune(4) | 27.6 | 9.9 | 29.5 | 23.8 | 22.7 |
| fine-tune(all) | 27.5 | 9.9 | 29.4 | 23.7 | **22.6** |
| LIN | 28.7 | 10.2 | 30.3 | 24.2 | 23.4 |
| LHN 1 | 28.4 | 10.2 | 30.5 | 24.4 | 23.4 |
| LHN 2 | 28.4 | 10.6 | 30.1 | 24.2 | 23.3 |
| LHN 3 | 28.0 | 10.3 | 30.0 | 24.0 | 23.1 |
| LHN 4 | 28.1 | 9.7 | 30.0 | 24.2 | **23.0** |
| LON | 28.1 | 10.2 | 30.1 | 24.4 | 23.2 |
| accent-specific | 26.4 | 8.8 | 28.7 | 22.5 | **21.6** |

# 4. Experiments and Results

## 4.1. Data and baseline model description

In our experiments, training set for the baseline model consists of 7000 hours of standard mandarin speech data. Adaptation set consists of 4 kinds of accents which are Anhui (AH), Beijing (BJ), Shanxi (SX) and Yunnan (YN) and each accent has 20 hours of data. Evaluation set consists of above 4 accents and additional 3 unseen accents which are Guizhou (GZ), Sichuan (SC) and Zhejiang (ZJ). Each accent in evaluation set has 2 hours of data. All the data is conversational telephone speech in real-world customer service scenario. 60-dimensional PLP features are used in the experiments.

All our ASR systems were implemented using the Kaldi toolkit [20]. Our baseline model to be adapted has 4 BLSTM layers and each layer has 512 hidden units for each direction. The targets are 5000 context-dependent tied phone states.

## 4.2. Comparison of selective fine-tuning and linear transformation

We first evaluated the performances of multi-accent adaptation using selective fine-tuning and linear transformation (LIN, LON and LHN).

For selective fine-tuning, we update the parameters of the first $n$ BLSTM layers, denoted as **fine-tune $(n)$**. The structure of our linear transformation follows the one in [21]. **LIN** is a linear transformation layer inserted after the input layer. LHN consists of two independent linear transformation layers inserted after the forward and backward direction of the $i_{th}$ hidden layer, denoted as **LHN $i$**. **LON** is a linear transformation layer inserted between the weight matrix of output layer and the softmax of the network. Weight matrixes of these linear transformation layers are initialized to the identity matrix and the bias vectors are initialized to zero.

WERs (word error rate) of 4 accents are shown in Table 1. **AVE** denotes average WER of all 4 accent evaluation sets. We mainly focus on the average WER of 4 accents, which roughly represents the performance of the system in the multi-accent applications. It turns out that multi-accent adaptation using fine-tuning all layers can get best performance. The performance on every accent except Beijing improves more or less. In terms of linear transformation, we find that LHN inserted after the

last hidden layer (LHN 4) is optimal, but is still worse than fine-tuning all layers. The reason why the linear transformation based methods don't work well is that the data used in the adaptation procedure is relatively more than the conventional problem the linear transformation is used to solve, such as speaker adaptation within a few utterances. With 80 hours of data, fine-tuning the entire model is relatively reliable and can adapt the model to multiple targets accents better.

In addition, we used the best method, fine-tuning all layers, to realize accent-specific adaptation. The hyperparameters were optimized for each model. The results are shown in the last row in Table 1. We notice that this method achieves 1% average absolute WER reduction compared with the multi-accent adaptation using fine-tuning all layers.

## 4.3. Improvements using gate unit

We next evaluated the performances of multi-accent adaptation using different gate units described in section 2.1. All gate units are inserted after the first hidden layer. Initializing all parameters in the gate unit randomly is shown to be optimal. Giving the parameters of the gate unit a bigger learning rate than the rest of network is also beneficial for the performance.

Table 2: *WERs (%) of models using different gate units*

| model | AH | BJ | SX | YN | AVE |
|---|---|---|---|---|---|
| baseline | 28.9 | 9.8 | 30.9 | 24.3 | 23.5 |
| GATE I | 27.1 | 9.0 | 28.7 | 22.9 | **21.9** |
| GATE II | 27.7 | 9.2 | 29.6 | 23.8 | 22.6 |
| GATE III | 28.1 | 10.4 | 30.1 | 24.7 | 23.3 |
| GATE IV | 27.0 | 9.2 | 29.3 | 23.6 | 22.3 |
| GATE V | 28.4 | 9.9 | 30.4 | 24.3 | 23.3 |

Table 2 gives the performances of different gate units. It turns out that the simplest gate GATE I performs best, which is contrast with the conclusion in [16]. The main reason is that in [16], the network with gate unit is trained form scratch with relatively more data so that the complex gate unit could be utilized well. However, in our approach, we add the gate unit on the basis of a well-trained baseline model. A complex gate unit would make the adaptation with limited data difficult to get a good performance.

Then we evaluated how many layers the gate unit applied to is best. We adjust the number of gate units by inserting them after the first $n$ hidden layers, denoted as **GATE I ($n$)**. The results are shown in Table 3. Inserting gate units after the first 3 layers is best in this experiment.

Table 3: *WERs (%) of models using different number of gate units*

| model | AH | BJ | SX | YN | AVE |
|---|---|---|---|---|---|
| baseline | 28.9 | 9.8 | 30.9 | 24.3 | 23.5 |
| GATE I (1) | 27.1 | 9.0 | 28.7 | 22.9 | 21.9 |
| GATE I (2) | 26.9 | 9.0 | 28.8 | 22.6 | 21.8 |
| GATE I (3) | 26.6 | 8.8 | 28.4 | 22.4 | **21.6** |
| GATE I (4) | 26.6 | 8.9 | 28.5 | 22.6 | 21.7 |

#### 4.4. Improvements using accent-specific top layer

Furthermore, we evaluated the performance of multi-accent adaptation using top layer. A bigger learning rate is adopted for the top layer, which make sense for that less data is fed into the accent-specific top layer than the bottom layers. We use learning rate factor (LR factor) to denote the ratio of the learning rate for accent-specific top layer to the one for bottom layers [22]. The performances of different LR factors are shown in Table 4. The best LR factor is shown to be 10.

Table 4: *WERs (%) of models using accent-specific top layer with different learning rate factors*

| model | AH | BJ | SX | YN | AVE |
|---|---|---|---|---|---|
| baseline | 28.9 | 9.8 | 30.9 | 24.3 | 23.5 |
| LR factor = 2 | 26.9 | 9.7 | 29.1 | 23.4 | 22.3 |
| LR factor = 5 | 26.8 | 9.6 | 29.0 | 23.2 | 22.2 |
| LR factor = 10 | 26.6 | 9.5 | 28.9 | 23.0 | **22.0** |
| LR factor = 20 | 26.6 | 9.4 | 29.5 | 23.3 | 22.2 |

#### 4.5. Improvements using accent-specific top layer with gate mechanism (AST-G)

Eventually we evaluated our proposed AST-G approach. We used the best configuration of the above two methods in this experiment. The results are shown in Table 5. The model using AST-G performs better than accent-specific adaptation on every target accent.

Table 5: *WERs (%) of the model using AST-G*

| model | AH | BJ | SX | YN | AVE |
|---|---|---|---|---|---|
| baseline | 28.9 | 9.8 | 30.9 | 24.3 | 23.5 |
| fine-tune(all) | 27.5 | 9.9 | 29.4 | 23.7 | 22.6 |
| accent-specific | 26.4 | 8.8 | 28.7 | 22.5 | 21.6 |
| **AST-G** | **26.1** | **8.7** | **28.0** | **22.1** | **21.2** |

#### 4.6. Improvements using multi-task learning with gate mechanism (MTL-G)

In this experiment, MTL-G uses 1 BLSTM layer and 1 DNN layer with sigmoid activation function at the top of the shared

bottom BLSTM layers as the secondary task network. The dimensionalities of these 2 layers are 512 and 256 respectively. The best gate unit GATE I is adopted. During adaptation, the shared bottom layers will be only updated using the gradient computed from the primary task network, which aims to avoid the disturbance from the secondary task network. The results are listed in Table 6. We find that MTL-G performs better than multi-accent adaptation using fine-tuning but worse than accent-specific adaptation. The performance degradation is tolerable because accent category labels won't be required for inference and it is very convenient in real-world applications.

Table 6: *WERs (%) of the model using MTL-G.*

| model | AH | BJ | SX | YN | AVE |
|---|---|---|---|---|---|
| baseline | 28.9 | 9.8 | 30.9 | 24.3 | 23.5 |
| fine-tune(all) | 27.5 | 9.9 | 29.4 | 23.7 | 22.6 |
| accent-specific | 26.4 | 8.8 | 28.7 | 22.5 | 21.6 |
| **MTL-G** | **27.3** | **9.7** | **28.9** | **23.4** | **22.3** |

Furthermore, we tested on 3 unseen accents consisting of Guizhou (GZ), Sichuan (SC) and Zhejiang (ZJ). The results are shown in Table 7. We surprisingly find that this approach can perform well on the unseen accents.

Table 7: *WERs (%) of the model using MTL-G on 3 unseen accents*

| model | GZ | SC | ZJ |
|---|---|---|---|
| baseline | 30.6 | 39.6 | 20.9 |
| **MTL-G** | **29.9** | **37.4** | **20.2** |

The performance improvements on target accents and unseen accents show that the multi-accent adaptation using MTL-G could lead to an integrated model which could perform well in multi-accent applications without the requirement of accent category labels for inference.

## 5. Conclusions

In this work, we proposed two multi-accent adaptation approaches to promote the performance of multi-accent speech recognition. AST-G uses accent-specific top layer and gate mechanism to model the distinctive characteristics of multiple target accents. This approach achieves 9.8% average relative WER reduction compared with the baseline model. Furthermore, to avoid the requirement of accent category labels, MTL-G jointly learns a multi-accent acoustic model and an accent classifier. In comparison with the baseline model, MTL-G achieves 5.1% average relative WER reduction.

For future work, we plan to extend MTL-G for its convenience in real-world applications.

## 6. Acknowledgements

# 7. References

[1] C. Huang, T. Chen, and E. Chang, "Accent issues in large vocabulary continuous speech recognition," *International Journal of Speech Technology*, vol. 7, no. 2-3, pp. 141–153, 2004.

[2] Y. Zheng, R. Sproat, L. Gu, I. Shafran, H. Zhou, Y. Su, D. Jurafsky, R. Starr, and S.-Y. Yoon, "Accent detection and speech recognition for shanghai-accented mandarin," in *Proc. Interspeech*, 2005.

[3] C. J. Leggetter and P. C. Woodland, "Maximum likelihood linear regression for speaker adaptation of continuous density hidden markov models," *Computer speech & language*, vol. 9, no. 2, pp. 171–185, 1995.

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

[5] G. Hinton, L. Deng, D. Yu, G. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, B. Kingsbury *et al.*, "Deep neural networks for acoustic modeling in speech recognition," *IEEE Signal processing magazine*, vol. 29, 2012.

[6] K. C. Sim, A. Narayanan, A. Misra, A. Tripathi, G. Pundak, T. N. Sainath, P. Haghani, B. Li, and M. Bacchiani, "Domain adaptation using factorized hidden layer for robust automatic speech recognition," in *Proc. Interspeech*, 2018, pp. 892–896.

[7] B. Li and K. C. Sim, "Comparison of discriminative input and output transformations for speaker adaptation in the hybrid nn/hmm systems," in *Proc. Interspeech*, 2010.

[8] R. Gemello, F. Mana, S. Scanzio, P. Laface, and R. De Mori, "Adaptation of hybrid ann/hmm models using linear hidden transformations and conservative training," in *Proc. ICASSP*. IEEE, 2006.

[9] H. Liao, "Speaker adaptation of context dependent deep neural networks," in *Proc. ICASSP*. IEEE, 2013, pp. 7947–7951.

[10] D. Yu, K. Yao, H. Su, G. Li, and F. Seide, "Kl-divergence regularized deep neural network adaptation for improved large vocabulary speech recognition," in *Proc. ICASSP*. IEEE, 2013, pp. 7893–7897.

[11] 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.

[12] L. Samarakoon and K. C. Sim, "Factorized hidden layer adaptation for deep neural network based acoustic modeling," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 12, pp. 2241–2250, 2016.

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

[14] 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 *Proc. ICASSP*. IEEE, 2013, pp. 7942–7946.

[15] Y. Huang, D. Yu, C. Liu, and Y. Gong, "Multi-accent deep neural network acoustic model with accent-specific top layer using the kld-regularized model adaptation," in *Proc. Interspeech*, 2014, pp. 2977–2981.

[16] S. Kim and M. L. Seltzer, "Towards language-universal end-to-end speech recognition," in *Proc. ICASSP*. IEEE, 2018, pp. 4914–4918.

[17] A. Jain, M. Upreti, and P. Jyothi, "Improved accented speech recognition using accent embeddings and multi-task learning," in *Proc. Interspeech*, 2018, pp. 2454–2458.

[18] H. Zheng, S. Zhang, L. Qiao, J. Li, and W. Liu, "Improving large vocabulary accented mandarin speech recognition with attribute-based i-vectors," in *Proc. Interspeech*, 2016, pp. 3454–3458.

[19] J.-T. Huang, J. Li, D. Yu, L. Deng, and Y. Gong, "Cross-language knowledge transfer using multilingual deep neural network with shared hidden layers," in *Proc. ICASSP*. IEEE, 2013, pp. 7304–7308.

[20] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz *et al.*, "The kaldi speech recognition toolkit," IEEE Signal Processing Society, Tech. Rep., 2011.

[21] M. Kitza, R. Schlüter, and H. Ney, "Comparison of blstm-layer-specific affine transformations for speaker adaptation," in *Proc. Interspeech*, 2018, pp. 877–881.

[22] P. Ghahremani, V. Manohar, H. Hadian, D. Povey, and S. Khudanpur, "Investigation of transfer learning for asr using lf-mmi trained neural networks," in *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2017, pp. 279–286.