Interspeech 2018
2-6 September 2018, Hyderabad

Factorized Deep Neural Network Adaptation for Automatic Scoring of L2 Speech in English Speaking Tests

Dean Luo[1], *Chunxiao Zhang*[1], *Linzhong Xia*[1], *Lixin Wang*[2]

[1] School of Electronic Communication Technology, Shenzhen Institute of Information Technology, China,
[2] Shenzhen Seaskyland Technologies, China
{luoda,zhangcx,xialz}@sziit.edu.cn, wlx@seaskylight.com

Abstract

Speaker adaptation has been shown to be effective on speech recognition and evaluation of L2 speech. However, other factors, such as environments and foreign accents, can affect the speech signal in addition to speakers. Factorizing the speaker, environment and other acoustic factors is crucial in evaluating L2 speech to effectively reduce acoustic mismatch between train and test conditions. In this study, we investigate the effects of deep neural network factorized adaptation techniques on L2 speech assessment in real speaking tests. Through recognition and automatic scoring experiments on L2 speech, we demonstrate that factorized fMLLR and iVector based DNN adaptation can better utilize adaptation data to efficiently adapt to complex speaker and environment conditions. Combining the factored components of iVectors and fMLLR transforms can further improve robustness of DNN models in speech recognition and automatic scoring of L2 speech in dynamic environments.

Index Terms: automatic scoring, acoustic factorization, speaker and environment adaptation, L2 speech assessment

1. Introduction

Automatic scoring techniques based on automatic speech recognition (ASR) have been developed to predict language learners' proficiency [1-3]. These techniques are usually based on posterior probabilities derived from ASR results of learners' L2 speech using acoustic models trained on native speech corpus. Because of acoustic mismatch between training and test conditions, recognition accuracy of L2 speech usually is much lower than L1 speech. In our previous studies, speaker adaptation techniques have been proposed for L2 speech evaluation [4,5]. These Maximum Likelihood Linear Regression (MLLR) based acoustic model adaptation techniques are developed for traditional Gaussian mixture model (GMM) ASR systems. Over the past few years, however, Deep Neural Networks (DNNs) have become the state-of-the-art in acoustic modeling, due to the improvement in accuracy over conventional HMM-GMM framework. We need to explore new adaptation techniques that can be applied to DNNs for L2 speech evaluation.

Speaker and environment adaptation techniques can fall into two categories: model adaption that modifies the parameters of the acoustic models to fit the test data and feature adaption that modifies features to better fit the trained models. Model adaption techniques such as MLLR and Maximum A Posteriori (MAP) for adapting GMMs cannot be applied to DNNs. Due to the significantly higher number of parameters in DNNs, it is hard to adapt DNNs with only a small amount of data. Feature adaption techniques such as fMLLR and i-vectors are used for adapting DNNs and show improvements over speaker-independent DNN models [6-8].

In this study, we investigate the effects of feature adaption techniques on DNN based automatic scoring of L2 English speech spoken by Chinese high school students. The data we used in our experiments are from the recordings of real English speaking tests using Seaskyland Technologies' E-exam system. In the Chinese city of Shenzhen, all high school students are required to take English speaking test twice a year, at the end of each school term. Most of them are taking the tests in a Computer-Assisted Language Learning (CALL) classroom with more than 40 students speaking simultaneously. Therefore the recording environments can be dynamic with unpredictable background noises. In addition to speaker adaption, environment adaptation is also important for our automatic scoring tasks.

In this paper, we investigate the effects of factored adaptation techniques on automatic scoring of L2 speech in speaking tests. Both fMMLR and iVector based factorization are examined and demonstrated to be effective on L2 speech recognition and assessment in complex environments.

2. Adaption Techniques

2.1. fMLLR adaptation

In the formulation of fMLLR, let $\mathbf{x}^{(t)}$ be the *d*-dimensional feature at time *t*, the adapted features are computed through the affine transform

$$\hat{\mathbf{x}}^{(t)} = A\mathbf{x}^{(t)} + \mathrm{b} = \mathbf{W}\boldsymbol{\xi}^{(t)}, \quad (1)$$

where $\mathbf{A}$ is a $d \times d$ square matrix, b is the $d \times 1$ bias term, $\boldsymbol{\xi} = [1 \ \mathbf{x}^T]^T$ is the input vector extended with an extra element equal to unity, and $\mathbf{W} = [\mathrm{b} \ \ \mathbf{A}]$ is the d by d+1 transformation matrix.

The transform parameters are estimated by optimizing the following auxiliary Q-function,

$$Q_{ML} = -\frac{1}{2}\sum_{t,m}\gamma_m(t)\{\log|A|^2 + \left(\mathbf{W}\boldsymbol{\xi}^{(t)} - \boldsymbol{\mu}^{(m)}\right)^T\boldsymbol{\Sigma}^{(m)-1}(\mathbf{W}\xi^{(t)} - \boldsymbol{\mu}^{(m)})\}, \quad (2)$$

where $\boldsymbol{\mu}^{(m)}$ and $\boldsymbol{\Sigma}^{(m)}$ are the mean and covariance for Gaussian component m and $\gamma_m(t)$ is the posterior probability of being in Gaussian m at time t.

If we assume the covariance matrices to be diagonal: $\boldsymbol{\Sigma}^{(m)} = diag([1/{\delta_1}^{(m)2} \ \ 1/{\delta_2}^{(m)2} \ \ \ldots \ 1/{\delta_n}^{(m)2}])$, let $\boldsymbol{w}_i$ be the transposed rows of $\mathbf{W}$, differentiating the auxiliary function with respect to the transform yields

$$\frac{\partial Q_{ML}}{\partial \boldsymbol{w}_i} = \beta\frac{\boldsymbol{p}_i}{\boldsymbol{p}_i{}^T\boldsymbol{w}_i} - \boldsymbol{G}^{(i)}\boldsymbol{w}_i + \boldsymbol{k}^{(i)} = 0, \quad (3)$$

10.21437/Interspeech.2018-2138

where $\beta = \sum_{t,m} \gamma_m(t)$, $\boldsymbol{p}_i$ is the extended vector cofactors of **A** and the sufficient statistics of $\boldsymbol{G}^{(i)}$ and $k^{(i)}$ are as follows:

$$\boldsymbol{G}^{(i)} = \sum_t \boldsymbol{\xi}^{(t)} \boldsymbol{\xi}^{(t)^T} \sum_m \frac{\gamma_m(t)}{\delta_i^{(m)2}} \quad (4)$$

$$k^{(i)} = \sum_t \boldsymbol{\xi}^{(t)} \sum_m \frac{\gamma_m(t)\mu_i^{(m)}}{\delta_i^{(m)2}} \quad (5)$$

The fMLLR transforms can be updated as

$$\boldsymbol{w}_i = \left(\alpha \boldsymbol{p}_i + \boldsymbol{k}^{(i)}\right) \boldsymbol{G}^{(i)-1} \quad (6)$$

where α satisfies

$$\alpha^2 \boldsymbol{p}_i{}^T \boldsymbol{G}^{(i)-1} \boldsymbol{p}_i + \alpha \boldsymbol{p}_i{}^T \boldsymbol{G}^{(i)-1} \boldsymbol{k}^{(i)} - \beta = 0 \quad (7)$$

2.2. Factorized fMLLR

[9] and [10] applied the nuisance attribute projection (NAP) [11] to acoustic latent factor analysis through orthogonal subspace projection. In this framework, a transform estimated in a complex acoustic environment can be projected onto speaker and environment subspaces. The two subspaces are constructed to be orthogonal so the factored transforms on different subspace are forced to be independent.

We use this approach to factorize fMMLR transforms. Consider an estimated d by d+1 fMLLR transformation matrix **W** described in **2.1.**, a super vector **w** can be constructed by stacking the columns of **W** into a single vector with the dimension of $\mathrm{D} = d(d+1)$. The training corpus can be represented by a matrix **E** whose columns are transform supervectors $\{\boldsymbol{w}_i\}$ that are estimated from the complete set of training data.

Following the latent factor analysis method of Kenny [12], the fMLLR transform supervector **w** which is dependent on speaker *s* and environment *e*, can be viewed as a sum of a speaker dependent vector and an environment dependent super vector, in addition to the sum of other components which approximates to the offset mean supervector $\bar{\boldsymbol{w}}$ over the entire training data:

$$\mathbf{w} \approx \bar{\boldsymbol{w}} + \mathbf{w}(\mathrm{s}) + \boldsymbol{w}(e) \quad (8)$$

To project **w** onto the two subspaces, a low-rank matrix **U** which is related to speaker variability and a matrix **V** which is related to environment variability are introduced to represent $\mathbf{w}(\mathrm{s})$ and $\boldsymbol{w}(e)$. The equation (8) becomes,

$$\mathbf{w} \approx \bar{\boldsymbol{w}} + [\mathbf{U}\ \mathbf{V}] \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{y} \end{bmatrix} \quad s.t.\ \boldsymbol{U} \perp \mathbf{V}$$

$$= \bar{\boldsymbol{w}} + \sum_{i=1}^{r_s} \boldsymbol{u}_i x_i + \sum_{j=1}^{r_e} \boldsymbol{v}_j y_j \quad (9)$$

where r_s is the number of speakers, r_e is the number of environments, **x** and **y** are factor-dependent weight vectors which quantify the amount of impact from speaker and environment, **U** is a $\boldsymbol{D} \times r_s$ matrix which represents speaker subspace and **V** is a $\mathbf{D} \times r_s$ matrix representing environment subspace.

In order to remove one direction, e.g., **V** which represents environment variability, a projection P can be defined:

$$\mathbf{P} = \mathbf{I} - \mathbf{V}(\boldsymbol{V}^T\boldsymbol{V})^{-1}\boldsymbol{V}^T \quad (10)$$

Operating projection **P** on (9) yields,

$$\mathbf{Pw} = \mathbf{P}\bar{\boldsymbol{w}} + \mathbf{PUx} + \mathbf{PVy}$$

$$= \mathbf{P}\bar{\mathbf{w}} + \mathbf{Ux} \quad (11)$$

The projection removes environment component $\mathbf{w}(\mathrm{e})$ in (8) leaving speaker component $\mathbf{w}(\mathrm{s})$ (i.e., $\mathbf{Ux}$) intact. With the idea of using Nuisance Attribute Projection (NAP) in [10], the design criterion for **P** and correspondingly **V** which can be viewed as a set of vectors $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_{r_e}]$, is

$$\mathbf{v}^* = \underset{\mathbf{v}, \|\mathbf{v}\|_2 = \mathbf{1}}{\operatorname{argmin}} \sum_{i,j} a_{i,j} \left\| \mathbf{P}(\mathbf{w}_i - \mathbf{w}_j) \right\|_2^2 \quad (12)$$

where w_i and w_j represent any pair of transforms in the training dataset E, $a_{i,j}$ represents weight parameters whose value are set to be 1 if $\mathbf{w}_i$ and $\mathbf{w}_j$ represent the transforms of the same speaker, and 0 otherwise.

The solution to (12) is an eigenvalue problem,

$$\mathbf{K}(diag(\mathbf{V1}) - \mathbf{V})\mathbf{Kv} = \lambda \mathbf{Kv} \quad (13)$$

where $\mathbf{K} = (\mathbf{PE})^{\mathrm{T}}(\mathbf{PE})$, and **1** is the vector of all ones. The detailed deviation can be found in [10].

2.3. iVector based factorized adaptation

iVectors which capture both speaker and environment specific information have been shown to be useful for rapid adaption of the neural network [6]. iVectors are fixed-dimensional representations, representing the coordinates, $\boldsymbol{\lambda}$ of a total variability subspace **M**. In iVector estimation, the difference between the means of a Gaussian Mixture Model (GMM) trained on all the data, μ_0 and speaker-specific means, $\boldsymbol{\mu}^s$, is assumed to be the matrix-vector product of the total variability matrix and the respective iVector:

$$\boldsymbol{\mu}^s = \boldsymbol{\mu}_0 + \boldsymbol{M}\boldsymbol{\lambda}^s \quad (14)$$

For adaptation, the iVectors are concatenated with the acoustic features, **x**. This produces bias adaptation of the first hidden layer, $\boldsymbol{h}$:

$$\boldsymbol{h} = \sigma(\mathbf{Wx} + \mathrm{b} + \mathbf{A}\boldsymbol{\lambda}), \quad (15)$$

where $\sigma(\cdot)$ denotes a nonlinearity, W and b are the corresponding weight matrix and bias, and $\mathbf{A}\boldsymbol{\lambda}$ is the bias contribution from the iVector with weight matrix **A**.

[18] used multi-condition training with neural networks to factorize speaker and environment information from iVectors. Specifically, this approach extracts bottleneck features from networks trained to classify either speakers or environments. The notion is that by learning to classify one factor, other nuisance factors are implicitly normalized out in the hidden representation since they are not relevant to classification task.

The procedure of factorizing iVectors can be summarized as follows: 1) Train independent multi-condition networks for speaker and environment classification using normal iVectors exacted from training data as inputs and the corresponding speaker or environment classes as outputs. 2) Extract bottleneck features with the same size of the input iVectors. 3) Substitute the original iVectors in acoustic features with extracted bottleneck features to train a DNN acoustic model. 4) At test time, extract normal iVectors from test data and pass them to exiting classification networks for factorization.

3. Experimental setup

3.1. Acoustic models

In DNN-HMM hybrid framework, an artificial neural network (ANN) is trained to output HMM context-dependent state-level probabilities [14]. The posteriors are converted into quasi-likelihoods by dividing by the prior of the states.

We trained HMM-GMM acoustic models with the Kaldi toolkit [15] and hybrid DNN models with nnet3 package using WSJ corpus [16] containing 282 speakers. Specifically, we trained monophone and triphone models on top of 13 mel-frequency cepstral coefficients (MFCCs) with delta and double deltas. We then trained on 40-dimensional features transformed with Linear Discriminant Analysis (LDA) and Maximum Likelihood Linear Transform (MLLT).

We trained 6-layer time-delay feedforward neural networks with p-norm activations (p=2) and input and output dimensions set to 2000 and 250, respectively. We trained with an

initial learning rate set to 0.005, which was reduced exponentially to a tenth of the original rate over 8 epochs. For baseline system, no fMLLR or iVector were used for adaptation. For iVector based adaptation, the iVectors were concatenated with the features for each frame using an iVector period of 10.

3.2. Automatic scoring

The confidence-based pronunciation assessment, which is defined as the Goodness of Pronunciation (GOP), is often used for assessing speakers' articulation and shows good results [2]. The GOP score is defined as follows.

$$\begin{aligned}\mathrm{GOP}(p) &= \log(\mathrm{p}(p|\mathbf{o})) \\ &\approx \log\frac{\mathrm{p}(\boldsymbol{o}|p)\mathrm{p}(p)}{\max\{q \in Q\}\, p(\boldsymbol{o}|q)p(q)} \\ &\approx \log\frac{\mathrm{p}(\boldsymbol{o}|p)}{\max\{q \in Q\}\, \mathrm{p}(\boldsymbol{o}|q)} \end{aligned} \quad (16)$$

where $\mathrm{p}(p|\mathbf{o})$ is the posterior probability that the speaker uttered phoneme p given speech observation $\mathbf{o}$, Q is the full set of phonemes. The numerator of (16) is a phone likelihood that can be calculated through phone-level HMM forced-alignment, and denominator is the likelihood of the phoneme recognized by HMM through a phone-loop network.

For DNN-HMM hybrid models, phone-level GOP can be implemented using the average frame posteriors which is the output of DNN [17]:

$$\mathrm{GOP}(p) = \mathrm{p}(\mathrm{p}|t_s, t_e; \boldsymbol{O}) = \frac{1}{t_e - t_s}\sum_{t_s}^{t_e} \mathrm{p}(s_t|\,\boldsymbol{o}_t), \quad (17)$$

where $p(s_t|\boldsymbol{o}_t)$ is the frame (state-level) posterior output of DNN, $\boldsymbol{o}_t$ is the feature observation at time t, t_s and t_e are the start and end time of phoneme p, which can be calculated through forced-alignment. We denote this implementation of GOP as GOP1.

As mentioned in 3.1, in DNN-HMM hybrid framework, the state-level posteriors can be converted to quasi-likelihoods by dividing by the prior of the states. Therefore, we can easily calculate the numerator and denominator likelihoods in (16) through forced-alignment:

$$\mathrm{p}(\boldsymbol{o}|p) \approx \tfrac{1}{t_e - t_s}\textstyle\sum_{t_s}^{t_e} \mathrm{p}(s_t|\,\boldsymbol{o}_t)/\mathrm{p}(s_t)\,, \quad (18)$$

$$p(\boldsymbol{o}|q) \approx \frac{1}{t_e - t_s}\sum_{t_s}^{t_e} \max\{s_t^* \in S\}(\,P(s_t^*|\,\boldsymbol{o}_t)/p(s_t^*)), \quad (19)$$

where S is the set of all states (or "senones"), $\mathrm{p}(s_t)$ are state-level priors calculated during DNN-HMM training. Applying (18) and (19) to (16), we implement another GOP score, denoted as GOP2. We consider GOP2 is more noise robust than GOP1, since the interfering factors due to acoustic mismatch appear in both the numerator and denominator of (16) and could be canceled out.

3.3. L2 speech data

The L2 speech data we used for our evaluation experiments are from recordings of reading-aloud section of Shenzhen senior high school English speaking tests over the period of 2014 to 2016. As mentioned in Introduction, high school students in Shenzhen are required to take English speaking test twice a year, at the end of each school term. In the reading-aloud section of the test, students are required to read-aloud the transcript of a one minute long video.

We selected 600 students (300 males and 300 females) with different levels of proficiency according to their official scores given by the teachers: 200 beginners, 200 intermedium learners and 200 advanced learners. For each speaker, there are 4 audio files which are their recordings of reading-aloud in 4 speaking tests at different times. Each audio file is associated with a label that records the speaker identity and the recording place (CALL classroom) ID, which can be used for speaker and environment labeling for adaptation.

Since there are 4 audio files with different environment IDs for each speaker, we randomly selected 3 of them as training data for adaptation, and the remaining 1 audio file for testing. Therefore the amount of adaptation data for each speaker is about 3 minutes, which is a reasonable amount of data for fMLLR or iVector adaption compared with other studies.

In order to evaluate the performance of automatic scoring, the 600 audio files in test data set were evaluated by 2 experts. Each expert gave an overall proficiency score from 0 (poorest) to 5 (highest) for each speaker. The correlation between the scores given by the two experts is 0.86. We used the average of the two expert scores for each speaker as the reference score to evaluate automatic scoring systems.

3.4. Adaptation setup

As mentioned in the previous section, there are 600 speakers in evaluation data. Considering 282 speakers in WSJ corpus, r_s in the equation (9) is 882. For environment variables, since the students were required to read aloud the transcript simultaneously, we considered the recordings in the same CALL classroom at the same time as one environment. The audio files were recorded in 12 different classrooms at 4 different times. If we include the clean recording environment of training corpus WSJ, the number of environments $r_e = 12 \times 4 + 1 = 49$.

For fMLLR adaptation, we constructed a global fMLLR transform for each speaker using KALDI in two ways: a) supervised estimation with reference transcript; b) unsupervised 2-pass estimation with the results of baseline ASR. The reason we performed unsupervised fMLLR adaptation is that there are pronunciation errors in adaptation data. We used ASR results of adaptation data with baseline DNN-HMM model and a phone-level bigram language model trained on transcripts instead of using forced-alignment of correct reference transcript. When all the speaker-level fMLLR transforms are estimated, we factorized them into speaker and environment transform vectors as described in section 3.3. We then applied factorized speaker transforms, denoted as ffM-s, or environment transforms, denoted as ffM-e, to the training features and retrain the neural networks. To jointly compensate the speaker and environment variabilities, the speaker transform ffM-s estimated for a speaker independently of environments (e.g. using adaption data of the same speaker in different environments), can be used in conjunction with the environment transform estimated for an environment independent of speaker identities. We denote this jointly applied transform as ffM-s+e.

The i-vectors are obtained as typical in Kaldi. As in [18], a single iVector was extracted across all the data for a given speaker or environment. For the bottleneck features, independent multi-condition networks were trained for speaker and environment classification, where each network had 882 or 49 output classes respectively. The networks have three 500-dimensional layers with the exception of middle bottleneck layers the size of the original iVectors. We denote factored bottleneck iVector features for speakers as bn-iv-s and for environments as bn-iv-e. We can concatenate bn-iv-s and bn-iv-e to form a new feature that combines speaker and environment factored bottleneck feature. We denote this feature as bn-iv-s+e.

Table 1: WER (%) results on L2 evaluation data with factored fMLLR adaptation and normal fMLLR.

baseline	fMLLR	ffM-s	ffM-e	ffM-s+e
17.27	16.24	16.35	15.50	15.33

Table 2: WER (%) results on L2 evaluation data with factored iVectors adaptation and normal iVector.

iVector	bn-iv-s	bn-iv-e	bn-iv-s+e	ffM-s+e, bn-iv-s+e
16.36	16.53	16.32	15.73	14.80

4. Experimental results and analysis

4.1. Recognition results

In order to evaluate the performance of the adapted models in speech recognition, we used utterances of 30 advanced learners whose scores given by the experts are the highest. The language model we used is a bigram model trained on all the transcripts of the evaluation data. Since these utterances contain very few pronunciation errors, we used supervised adaptation for fMLLR.

The results of factored fMLLR adaptation are shown in Table 1. ffM-e and ffM-s+e outperform speaker-level fMLLR. We consider it is because the recording environment of the test data of a speaker was always different than those in the training set, the mismatch caused by environment cannot be effectively compensated by the normal fMLLR. ffM-e, however, can learn from more data of the same environment (e.g. other students' utterances recorded in the same CALL classroom). ffM-s+e shows the best performance further confirms the effectiveness of factored adaption on speech in dynamic environments.

As shown in Table 2, iVector-based factored adaptation also shows the same trend: factored component of environment specific adaptation is more effective than normal joint iVector or speaker specific bn-iv-s. Combining bn-iv-s+e with ffM-s+e yields lowest WER, which indicates that the two factorized adaption techniques can work together for best performance.

4.2. Automatic scoring results

The correlations between automatic scores and reference scores with supervised and unsupervised fMLLR based factored adaptation are shown in Table 3 and 4. As expected, GOP2 shows better performance with all models than GOP1. Unsupervised fMLLR based models show better performances than supervised adaptation. This indicates that unlike in the tasks of recognition where supervised adaptation generally shows better performance, in the case of automatic scoring for L2 speech, unsupervised 2-pass adaptation can yield better results. Similar to the recognition experiments, ffM-e+s shows best performances of other fMLLR based adapted models.

The automatic scoring results with iVector based adaptation are shown in Table 5. Although the results with iVectors seem slightly worse than those with fMLLR based adaptation, bn-iv-s+e outperforms the normal iVector, bn-iv-s and bn-iv-e adaptation. Combining ffM-s+e with bn-iv-s+e can further improve the performance and yields the highest correlation of 0.86. This indicates that both iVector based and fMLLR based factored adaptations are more effective than the normal adaption with a single transform for all components. Combining factored iVectors with fMLLR factored transforms can further improve automatic scoring performance.

Table 3: Correlations between automatic scores and reference scores with factored supervised fMLLR adaptation compared with baseline (no adaption)

scheme	base-line	fMLLR	ffM-s	ffM-e	ffM-s+e
GOP1	0.70	0.72	0.72	0.77	0.77
GOP2	0.77	0.80	0.81	0.82	0.83

Table 4: Correlations between automatic scores and reference scores with factored unsupervised fMLLR adaptation compared with baseline (no adaption)

scheme	base-line	fMLLR	ffM-s	ffM-e	ffM-s+e
GOP1	0.70	0.75	0.76	0.79	0.79
GOP2	0.77	0.80	0.82	0.83	0.85

Table 5: Correlations between automatic scores and reference scores with factored iVector adaptation

scheme	iVec-tor	bn-iv-s	bn-iv-e	bn-iv-s+e	ffM-s+e,bn-iv-s+e
GOP1	0.71	0.72	0.73	0.75	0.77
GOP2	0.77	0.79	0.80	0.82	0.86

5. Conclusions

We investigated the effects of factored adaptation of DNN acoustic models on automatic scoring of L2 speech in real speaking tests. Experimental results show that factorized adaptation techniques can utilize speaker or environment features more efficiently, especially in the case that involves multiple speakers speaking in complex environments. Combining the factored components of iVectors and fMLLR transforms can further improve robustness of DNN model in speech recognition and automatic scoring of L2 speech in dynamic environments.

Future works include acoustic modelling of L2 speech with more data and exploring other adaptation or transfer learning techniques for L2 speech recognition and assessment.

6. Acknowledgement

This work is supported jointly by the projects of Shenzhen science and technology innovation committee (GRCK2017042409560810, GRCK2017042409552883 and JCYJ20170817114522834), Shenzhen Institute of Information Technology research fund (PT201701), the Guangdong Province higher vocational colleges & schools Pearl River scholar funded scheme (2016) and the Major Program of the National Social Science Fund of China 13&ZD189.

7. References

[1] Trofimovich, P., & Baker, W. (2006). Learning second language suprasegmentals: Effect of L2 experience on prosody and fluency characteristics of L2 speech. *Studies in Second Language Acquisition*, 28, 1–30.

[2] S.M. Witt and S.J. Young.(2000). Phone-level Pronunciation Scoring and Assessment for Interactive Language Learning,"Speech Communications, 30 (2–3): pp.95-108.

[3] Tsurutani, C. (2010). Foreign accent matters most when timing is wrong, *Interspeech 2010* 1854-57.

[4] Dean Luo, Yu Qiao, Nobuaki Minematsu, Yutaka Yamauchi, Keikichi Hirose:Regularized-MLLR speaker adaptation for computer-assisted language learning system. INTERSPEECH 2010: 594-597.
[5] Dean Luo, Yu Qiao, Nobuaki Minematsu, Yutaka Yamauchi, Keikichi Hirose: Analysis and utilization of MLLR speaker adaptation technique for learners' pronunciation evaluation. INTERSPEECH 2009: 608-611
[6] G. Saon, H. Soltau, D. Nahamoo, and M. Picheny, "Speaker adaptation of neural network acoustic models using i-vectors," in In Proceedings of IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU), 2013.
[7] Y. Miao, H. Zhang, and F. Metze, "Speaker adaptive training of deep neural network acoustic models using i-vectors," IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 23, no. 11, pp. 1938–1949, 2015.
[8] F. Seide, G. Li, X. Chen, and D. Yu, "Feature engineering in context dependent deep neural networks for conversational speech transcription," in In Proceedings of IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU), 2011..
[9] Seo, Hyunson, H. G. Kang, and M. L. Seltzer. "Factored adaptation of speaker and environment using orthogonal subspace transforms." IEEE International Conference on Acoustics, Speech and Signal Processing IEEE, 2014:3251-3255.
[10] Solomonoff, A., C. Quillen, and W. M. Campbell. "Channel Compensation for SVM Speaker Recognition." Acoustics, Speech, and Signal Processing, 2005. Proceedings. (ICASSP '05). IEEE International Conference on IEEE, 2013:629-632.
[11] W. Campbell, D. Sturim, D. Reynolds, and A. Solomonoff,"SVM based speaker verification using a GMM supervector kernel and NAP variability compensation," in Proc. ICASSP, 2006
[12] P. Kenny and P. Dumouchel, "Experiments in speaker verification using factor analysis likelihood ratios," in Proc. Odyssey04, 2004, pp. 219-226.
[13] W. Mattheyses, W. Verhelst, and P. Verhoeve (2006). Robust pitch marking for prosodic modification of speech using td-psola, in Proceedings of the 2nd Annual IEEE Benelux/DSP Valley Signal Processing Symposium (SPS-DARTS '06), pp.43–46.
[14] X. Zhang, J. Trmal, D. Povey, and S. Khudanpur, "Improving deep neural network acoustic models using generalized maxout networks," in Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on. IEEE, 2014, pp. 215–219.
[15] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, "The Kaldi speech recognition toolkit," in Proc. ASRU, 2011.
[16] J. Garofalo, D. Graff, D. Paul, and D. Pallett, "CSR-I (WSJ0) Complete LDC93S6A," Linguistic Data Consortium, Philadelphia, 2007
[17] W. Hu, et al.(2013), A New DNN-based High Quality Pronunciation Evaluation for Computer-Aided Language Learning (CALL) , Proc. INTERSPEECH 2013, 1886-1890
[18] Fainberg, Joachim, S. Renals, and P. Bell. "Factorised Representations for Neural Network Adaptation to Diverse Acoustic Environments." INTERSPEECH 2017:749-753.