CSLT TECHNICAL REPORT-20160014 [Saturday 7th January, 2017]

Exploring The Role of Deep Speaker Features for Speaker Verification

Lantian Li[1,2]
and Dong Wang[1,3*]

*Correspondence: wang-dong99@mails.tsinghua.edu.cn
[1]Center for Speech and Language Technology, Research Institute of Information Technology, Tsinghua University, ROOM 1-303, BLDG FIT, 100084 Beijing, China
Full list of author information is available at the end of the article

Abstract

Recent research shows that deep neural networks (DNNs) can be used to extract deep speaker features that preserve speaker characteristics and can be used in speaker verification. In this paper, we analyse the role of deep speaker features extracted from different DNN layers in speaker verification task.

Keywords: deep neural networks; speaker vector; speaker verification

1 Introduction

Speaker verification, also known as voiceprint recognition, is an important biometric authentication technology that has been widely used to verify speakers' identities. According to the words that are allowed to speak in enrollment and test, speaker verification system can be categorized into either text-dependent or text-independent. While a text-dependent system requires the same words/sentences to be spoken in test and enrollment, a text-independent system permits any words to speakers. This paper focuses on a semi text-independent scenario where the words for enrollment and test are constrained in a limited set of short phrases, e.g., 'turn on the radio'. With this limitation, people can speak different sentences in enrollment and test while the system performance keeping not deteriorated, which makes the system more acceptable in practice.

Most of the successful approaches to speaker verification are based on generative models and with unsupervised learning, e.g., the famous Gaussian mixture model-universal background model (GMM-UBM) framework [1]. A number of advanced models have been proposed based on the GMM-UBM architecture. The i-vector model [2] [3] is among the most successful. Despite the impressive success, the GMM-UBM model and the subsequent i-vector model share the intrinsic disadvantage of all unsupervised learning methods: the goal of the model training is to describe the distributions of acoustic features, instead of discriminating speakers.

This problem can be solved in two directions. The first direction is to employ various discriminative models to enhance the generative framework. For example, the SVM model for GMM-UBMs [4], and the PLDA model for i-vectors [5]. All these approaches provide significant improvement over the baseline. Another direction is to look for more discriminative features, i.e., the features that are more sensitive to speaker change and less sensitive to change of other irrelevant factors, such as phone contents and channels [6]. However, the improvement obtained by the 'feature engineering' is much less significant compared to the achievements obtained by the

discriminative models such as SVM and PLDA. A possible reason is that most of the features are human-crafted and thus tend to be suboptimal in practical usage.

Recent research on deep learning offers a new idea of 'feature learning'. It has been shown that with a deep neural network (DNN), task-oriented features can be learned layer by layer from very raw features. For example in automatic speech recognition (ASR), phone-discriminative features can be learned from spectrum or filter bank energies (Fbanks). This learned features are very powerful and have defeated the Mel frequency cepstral coefficient (MFCC) feature that has dominated in ASR for several decades [7].

This favorable property of DNNs in learning task-oriented features can be utilized to learn speaker-related features as well. A recent study shows that this is possible at least in text-dependent tasks [8]. The authors constructed a DNN model and set the training objective as to discriminate a set of speakers, and for each frame, the speaker-related features were read from the activations of the last hidden layer. They tested the method on a foot-print text-dependent speaker verification task (only a short phrase 'ok, google'). The experimental results showed that reasonable performance can be achieved with the DNN-based features, although it is still difficult to compete with the i-vector baseline.

In this paper, we extend the application of the DNN-based feature learning approach to semi text-independent tasks, and present a phone-dependent training which involves phone posteriors obtained from an ASR system in the training. The experimental results show that the DNN-based feature learning works well on text-independent tasks, actually even better than on text-dependent tasks, and the phone-dependent training offers marginal but consistent gains.

The rest of this paper is organized as follows. Section 2 describes the related work, and Section 3 presents the DNN-based speaker feature learning. The experiments are presented in Section 4, and Section **??** concludes the paper.

2 Related work

This paper follows the work in [8]. The difference is that we extend the application of the DNN-based feature learning approach to semi text-independent tasks, and we introduce a phone-dependent training. Due to the mismatched content of the enrollment and test speech, our task is more challenging.

The DNN model has been employed in speaker verification in other ways. For example, in [9], DNNs trained for ASR were used to replace the UBM model to derive the acoustic statistics for i-vector model training. In [10], a DNN was used to replace PLDA to improve discriminative capability of i-vectors. All these methods rely on the generative framework, i.e., the i-vector model. The DNN-based feature learning presented in this paper is purely discriminative, without any generative model involved.

3 DNN-based feature learning

This section presents the DNN-based feature learning. We first describe the main structure of the model and the learning process, and propose the phone-dependent learning. Finally the difference between the i-vector approach and the DNN-based approach is discussed.

3.1 DNN-based feature extraction

It is well-known that DNNs can learn task-oriented features from raw features layer by layer. This property has been employed in ASR where phone-discriminative features are learned from very low-level features such as Fbanks or even spectrum [7]. It has been shown that with a well-trained DNN, variations irrelevant to the learning task are gradually eliminated when the input feature is propagated through the DNN structure layer by layer. This feature learning is so powerful that in ASR, the primary Fbank feature has defeated the MFCC feature that has been carefully designed by people and dominated in ASR for several decades.

This property can be also employed to learn speaker-related features. Actually researchers have put much effort in looking for features that are more discriminative for speakers [6], but the effort is mostly vain and the MFCC is still the most popular choice. The success of DNNs in ASR suggests a new direction that speaker-related features can be learned from data instead of crafted by hand. The learning can be easily done and the process is rather similar as in ASR, with the only difference that in speaker verification, the learning goal is to discriminate different speakers.

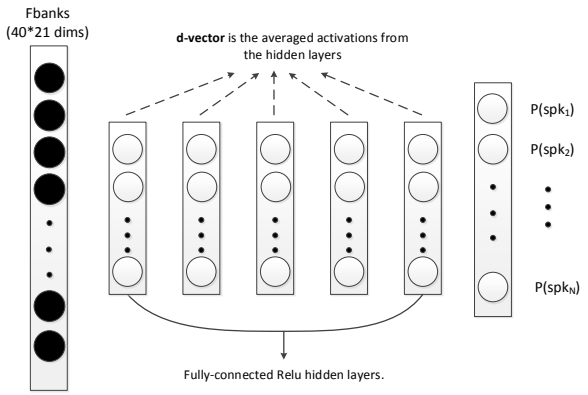

Figure 1 *The DNN structure used for learning speaker-related features.*

Figure 1 presents the DNN structure used for the speaker-related feature learning. Following the convention of ASR, 40-dimensional Fbanks are extracted from each frame and 21 frames are stacked together (10 frames for left and right context) as the DNN input. There are 5 hidden layers, and each consists of 256 units. The units of the output layer correspond to the speakers in the training data. The 1-hot encoding scheme is used to label the target, and the training criterion is set to cross entropy.

Once the DNN has been trained successfully, the speaker-related features can be read from any hidden layer. The features are extracted for all the frames of the given utterance, and the features are averaged to form a speaker vector. Following the nomenclature in [8], we call this speaker vector as 'd-vector'. Similar to i-vectors, a d-vector represents the speaker identity of an utterance in the speaker space. The same methods used for i-vectors can be used for d-vectors to conduct the test, for example by computing the cosine distance or applying PLDA. In this paper, we explore that features extracted from which hidden layer are more robustness and generalization.

3.2 Comparison between i-vectors and d-vectors

The two kinds of speaker vectors, the d-vector and the i-vector, are fundamentally different. I-vectors are based on a linear Gaussian model, for which the learning is unsupervised and the learning criterion is maximum likelihood on acoustic features. In contrast, d-vectors are based on neural networks, for which the learning is supervised, and the learning criterion is maximum discrimination for speakers. This difference in model structures and learning methods leads to significant different properties of these two vectors.

First, the i-vector is 'descriptive', which represents the speaker by constructing a GMM (derived from the i-vector) to fit the acoustic features. In contrast, the d-vector is 'discriminative', which represents the speaker by removing speaker-irrelevant variance.

Second, the i-vector can be regarded as a 'global' speaker description, which is inferred from 'all' the frames of an utterance; however the d-vector is a 'local' description, which is inferred from 'each' frame, and only the context information is used in the inference. This means that the d-vector tends to be more superior with a short utterance, while the i-vector tends to perform better with a relative long utterance.

Third, the i-vector approach more relies on the enrollment data to form a reasonable distribution that can be used to discriminate different speakers; whereas the d-vector approach more relies on the 'universal' data to learn speaker-related features. This means that a large amount of training data (labelled with speakers) is much more important and useful for the d-vector approach.

4 Experiments

4.1 Database

- **Development sets**:
- WSJ-DEV: 200 speakers with $24,031$ utterances are randomly selected from the WSJ database to train the i-vector, LDA and PLDA models. The same data is also used to conduct the deep speaker feature learning.
- CSLT-C300-DEV: 200 female speakers with $20,000$ utterances are used as the development set similar as WSJ-DEV.
- **Evaluation set**:
- WSJ-EVA: The evaluation set consists of 110 enrollment utterances and $14,336$ test utterances, and it is based on the pair-wised $1,455,960$ trials, including $13,236$ target trials and $1,442,724$ imposter trials.
- CSLT-C300-EVA: The evaluation set consists of 100 enrollment utterances and $10,000$ test utterances, and it is based on the pair-wised $900,000$ trials, including $9,000$ target trials and $891,000$ imposter trials.

4.2 Experimental results

4.2.1 Baseline of i-vector model with MFCCs

We first present the basic results obtained with i-vector models with cosine scoring(Cosine), LDA and PLDA. The dimension of the i-vector is fixed to 400, and the number of Gaussian components is set to $2,048$. Again, the dimension of the LDA projection space is set to 150.

	Chi	Eng	Chi-Eng	Eng-Chi	ALL-Chi	ALL-Eng
Cosine	1.32	0.94	10.30	4.27	1.09	0.82
LDA	1.13	0.62	4.91	3.44	0.89	0.69
PLDA	1.00	0.60	2.08	2.31	0.81	0.60
LDA-PLDA	1.04	0.65	2.08	2.51	0.93	0.65

Table 1 *EER(%) results of* ***MFCCs*** *on the* ***i-vector*** *systems.*

4.2.2 d-vector baseline

This experiment examines the d-vector approach on the text-independent task. The d-vectors are extracted from varying hidden layers, and we attempt to explore the effect of varying hidden layer position (1 to 5, respectively).

	Chi	Eng	Chi-Eng	Eng-Chi	ALL-Chi	ALL-Eng
Cosine	2.81	1.73	3.34	4.44	3.56	2.13
LDA	1.64	0.97	5.02	4.06	1.63	0.97
PLDA	4.67	3.11	4.31	5.14	5.98	3.00
LDA-PLDA	1.44	0.87	1.89	3.13	1.42	0.85

Table 2 *EER(%) results of* 1^{th}*-layer on the* ***d-vector*** *systems.*

	Chi	Eng	Chi-Eng	Eng-Chi	ALL-Chi	ALL-Eng
Cosine	2.66	1.56	3.56	4.18	3.18	1.84
LDA	1.50	0.96	5.35	3.87	1.53	0.82
PLDA	3.89	2.43	3.32	4.43	5.06	2.34
LDA-PLDA	1.23	0.87	2.18	3.00	1.38	0.81

Table 3 *EER(%) results of* 2^{th}*-layer on the* ***d-vector*** *systems.*

4.2.3 d-vector with bottleneck features

	Chi	Eng	Chi-Eng	Eng-Chi	ALL-Chi	ALL-Eng
Cosine	2.58	1.67	3.51	4.06	2.79	1.63
LDA	1.36	0.88	5.27	4.62	1.44	0.90
PLDA	3.24	2.37	3.42	3.96	4.51	2.06
LDA-PLDA	1.13	0.85	2.49	4.16	1.36	0.88

Table 4 *EER(%) results of 3^{th}-layer on the* ***d-vector*** *systems.*

	Chi	Eng	Chi-Eng	Eng-Chi	ALL-Chi	ALL-Eng
Cosine	2.57	2.24	3.43	4.08	2.87	2.08
LDA	1.38	0.95	4.80	4.26	1.31	0.93
PLDA	3.70	2.74	3.82	4.34	4.19	2.15
LDA-PLDA	1.17	0.94	2.61	3.72	1.20	1.05

Table 5 *EER(%) results of 4^{th}-layer on the* ***d-vector*** *systems.*

	Chi	Eng	Chi-Eng	Eng-Chi	ALL-Chi	ALL-Eng
Cosine	4.90	4.05	3.81	4.64	5.42	3.87
LDA	1.26	1.08	5.11	4.90	1.20	1.10
PLDA	5.57	4.90	3.86	4.77	6.88	3.11
LDA-PLDA	1.18	1.15	2.98	4.40	1.19	1.32

Table 6 *EER(%) results of 5^{th}-layer on the* ***d-vector*** *systems.*

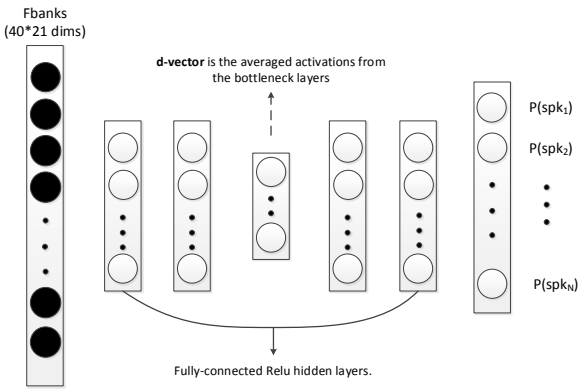

Figure 2 *Deep speaker features extraction with the bottleneck structure.*

	Chi	Eng	Chi-Eng	Eng-Chi	ALL-Chi	ALL-Eng
Cosine	3.88	2.09	2.92	4.44	4.41	2.16
LDA-30	2.41	1.43	5.26	6.13	2.74	1.58
PLDA	29.46	12.21	33.51	16.38	27.23	18.54
LDA-PLDA	2.30	1.42	2.28	5.16	2.52	1.53

Table 7 *EER(%) results of 1^{th}-layer* ***bottleneck*** *on the* ***d-vector*** *systems.*

	Chi	Eng	Chi-Eng	Eng-Chi	ALL-Chi	ALL-Eng
Cosine	2.83	1.64	2.79	3.88	3.20	1.69
LDA-30	1.97	1.06	5.98	5.22	2.22	1.15
PLDA	22.18	16.00	15.10	18.57	26.28	15.80
LDA-PLDA	1.86	1.13	2.63	4.70	2.48	1.42

Table 8 *EER(%) results of 2^{th}-layer* ***bottleneck*** *on the* ***d-vector*** *systems.*

	Chi	Eng	Chi-Eng	Eng-Chi	ALL-Chi	ALL-Eng
Cosine	2.42	1.46	3.23	4.11	2.71	1.52
LDA-30	1.79	1.00	6.58	4.66	2.18	1.02
PLDA	16.31	13.08	21.72	18.97	16.62	9.65
LDA-PLDA	1.82	1.07	3.13	4.30	2.23	1.27

Table 9 *EER(%) results of 3^{th}-layer* ***bottleneck*** *on the* ***d-vector*** *systems.*

	Chi	Eng	Chi-Eng	Eng-Chi	ALL-Chi	ALL-Eng
Cosine	2.37	1.86	3.52	4.94	2.74	1.89
LDA-30	1.69	1.25	7.96	6.17	1.97	1.22
PLDA	12.57	8.60	19.14	14.80	16.11	9.44
LDA-PLDA	1.69	1.48	3.33	6.11	2.40	1.45

Table 10 *EER(%) results of 4^{th}-layer **bottleneck** on the **d-vector** systems.*

	Chi	Eng	Chi-Eng	Eng-Chi	ALL-Chi	ALL-Eng
Cosine	3.07	3.12	4.21	5.87	3.00	3.02
LDA-30	2.00	1.82	7.83	7.11	2.50	2.00
PLDA	7.13	4.34	7.55	6.69	5.56	4.93
LDA-PLDA	2.00	1.96	3.60	8.30	2.63	2.15

Table 11 *EER(%) results of 5^{th}-layer **bottleneck** on the **d-vector** systems.*

4.2.4 i-vector with bottleneck features

	Chi	Eng	Chi-Eng	Eng-Chi	ALL-Chi	ALL-Eng
Cosine	9.73	4.31	31.11	16.68	6.11	3.51
LDA-150	8.70	4.47	26.28	17.32	6.06	3.74
PLDA	7.80	4.00	9.97	12.56	5.13	3.13
LDA-PLDA	7.54	3.71	9.15	12.70	5.48	3.13

Table 12 *EER(%) results of 1^{th}-layer **bottleneck** on the **i-vector** systems.*

	Chi	Eng	Chi-Eng	Eng-Chi	ALL-Chi	ALL-Eng
Cosine	7.67	3.44	24.56	15.81	4.24	2.95
LDA-150	6.72	3.36	21.21	14.59	4.02	3.20
PLDA	5.76	2.61	9.26	8.50	3.28	2.39
LDA-PLDA	5.39	2.49	8.73	8.27	3.44	2.37

Table 13 *EER(%) results of 2^{th}-layer **bottleneck** on the **i-vector** systems.*

	Chi	Eng	Chi-Eng	Eng-Chi	ALL-Chi	ALL-Eng
Cosine	5.68	2.83	28.00	16.40	3.72	2.64
LDA-150	5.09	2.78	22.64	15.58	3.64	2.83
PLDA	4.17	2.20	7.86	7.52	2.86	2.17
LDA-PLDA	3.97	2.04	7.31	7.57	2.81	2.08

Table 14 *EER(%) results of 3^{th}-layer **bottleneck** on the **i-vector** systems.*

4.2.5 i-vector with BN + MFCC features

	Chi	Eng	Chi-Eng	Eng-Chi	ALL-Chi	ALL-Eng
Cosine	4.58	2.84	23.94	14.74	3.07	2.60
LDA-150	3.86	2.92	18.70	14.18	2.97	3.10
PLDA	3.29	2.21	7.84	8.58	2.18	2.12
LDA-PLDA	3.07	2.12	7.31	8.71	2.33	2.16

Table 15 *EER(%) results of 4^{th}-layer* ***bottleneck*** *on the* ***i-vector*** *systems.*

	Chi	Eng	Chi-Eng	Eng-Chi	ALL-Chi	ALL-Eng
Cosine	5.58	4.36	25.45	13.10	3.32	3.33
LDA-150	4.66	4.03	21.92	15.72	3.47	3.85
PLDA	4.08	3.29	8.77	12.29	3.04	3.11
LDA-PLDA	3.97	3.29	8.71	12.86	3.22	3.09

Table 16 *EER(%) results of 5^{th}-layer* ***bottleneck*** *on the* ***i-vector*** *systems.*

	Chi	Eng	Chi-Eng	Eng-Chi	ALL-Chi	ALL-Eng
Cosine	2.92	1.38	23.20	8.43	1.67	1.19
LDA-150	2.44	1.07	13.01	7.49	1.47	1.09
PLDA	1.74	0.78	3.16	4.03	1.07	0.80
LDA-PLDA	1.83	0.78	2.98	3.96	1.20	0.79

Table 17 *EER(%) results of 1^{th}-layer* ***BN + MFCC*** *on the* ***i-vector*** *systems.*

	Chi	Eng	Chi-Eng	Eng-Chi	ALL-Chi	ALL-Eng
Cosine	2.52	1.03	20.44	7.39	1.33	1.00
LDA-150	1.98	0.82	9.97	5.51	1.19	0.91
PLDA	1.38	0.62	3.23	3.32	0.84	0.72
LDA-PLDA	1.41	0.69	3.17	3.46	1.06	0.73

Table 18 *EER(%) results of 2^{th}-layer* ***BN + MFCC*** *on the* ***i-vector*** *systems.*

	Chi	Eng	Chi-Eng	Eng-Chi	ALL-Chi	ALL-Eng
Cosine	2.39	1.11	21.83	7.28	1.30	1.02
LDA-150	1.81	0.85	10.99	5.60	1.11	0.93
PLDA	1.19	0.70	3.33	3.30	0.81	0.74
LDA-PLDA	1.26	0.78	3.30	3.60	0.96	0.77

Table 19 *EER(%) results of 3^{th}-layer* ***BN + MFCC*** *on the* ***i-vector*** *systems.*

	Chi	Eng	Chi-Eng	Eng-Chi	ALL-Chi	ALL-Eng
Cosine	2.16	1.20	21.60	7.64	1.23	1.04
LDA-150	1.74	0.95	10.89	5.96	1.19	1.00
PLDA	1.26	0.77	3.60	3.77	0.68	0.70
LDA-PLDA	1.29	0.82	3.61	4.00	0.84	0.74

Table 20 *EER(%) results of 4^{th}-layer* ***BN + MFCC*** *on the* ***i-vector*** *systems.*

	Chi	Eng	Chi-Eng	Eng-Chi	ALL-Chi	ALL-Eng
Cosine	2.88	1.51	22.67	8.92	1.32	1.31
LDA-150	2.21	1.23	12.36	10.38	1.39	1.25
PLDA	1.74	0.99	4.35	5.29	1.00	0.97
LDA-PLDA	1.72	1.05	4.21	5.81	1.21	0.99

Table 21 *EER(%) results of 5^{th}-layer* ***BN + MFCC*** *on the* ***i-vector*** *systems.*

5 Conclusions

This paper investigated the DNN-based feature leaning for speaker recognition, and studied the performance of this approach on a semi text-independent speaker verification task. The experimental results demonstrated that this approach (d-vectors) can offer reasonable performance, and outperformed the i-vector baseline with simple cosine distance. However, when discriminative normalization methods such as LDA and PLDA are applied, the i-vector approach exhibits better performance.

Although it has not beat the i-vector approach at present, the d-vector approach is quite promising. We argue that an obvious advantage of the i-vector system is that it smartly combines the power of generative models (GMM) and discriminative models (LDA, PLDA), which the current d-vector approach has to learn. Nevertheless, as has been demonstrated in this paper, the d-vector approach is potential in learning speaker-related features with large amounts of universal data, which is a big advantage compared to the i-vector approach for which the universal data is used for inferring the speaker space only. Another merit with the d-vector approach is that the local learning property, which enables speaker characters being identified with very short utterances. This is impossible for the i-vector approach which requires much more data to infer the speaker characters.

The future work involves investigating strong statistical models for d-vectors. The current average-based accumulation is too simple to model the statistical property of speakers' behavior, which is a major shortage compared to the i-vector model. Another work is to utilize more universal data to learn speaker-related features, and test on large scale text-independent tasks.

6 Experimental analysis

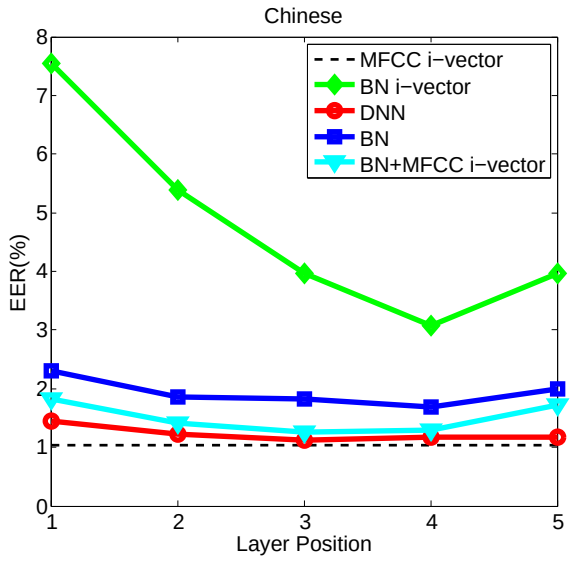

Figure 3 *The effect of varying layer position in Chinese test condition.*

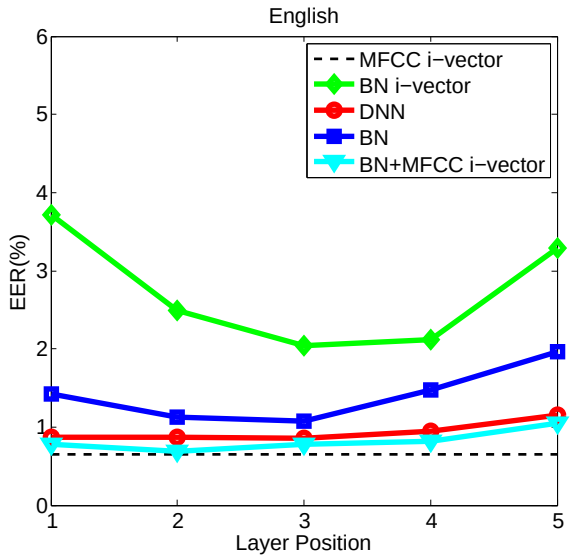

Figure 4 *The effect of varying layer position in English test condition.*

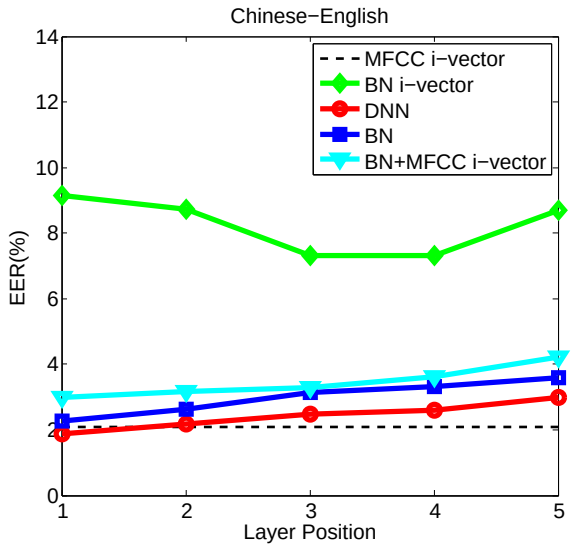

Figure 5 *The effect of varying layer position in Chinese-English test condition.*

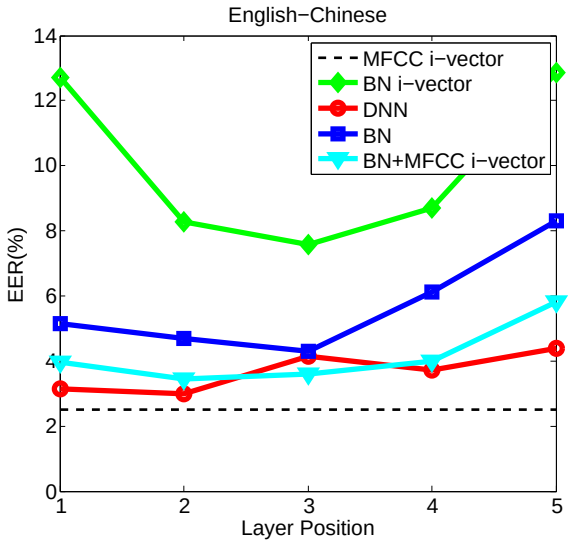

Figure 6 *The effect of varying layer position in English-Chinese test condition.*

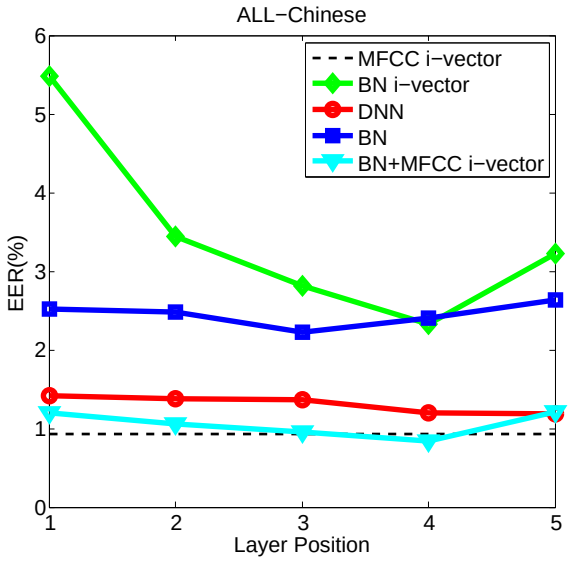

Figure 7 *The effect of varying layer position in ALL-Chinese test condition.*

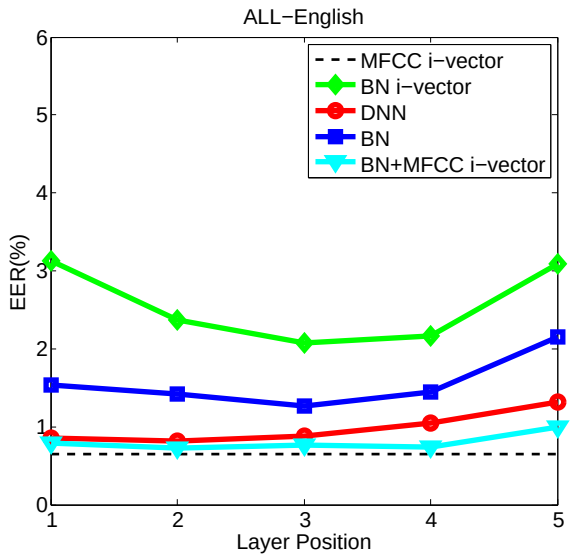

Figure 8 *The effect of varying layer position in ALL-English test condition.*

Acknowledgement

Author details
[1]Center for Speech and Language Technology, Research Institute of Information Technology, Tsinghua University, ROOM 1-303, BLDG FIT, 100084 Beijing, China. [2]Center for Speech and Language Technologies, Division of Technical Innovation and Development, Tsinghua National Laboratory for Information Science and Technology, ROOM 1-303, BLDG FIT, 100084 Beijing, China. [3]Department of Computer Science and Technology, Tsinghua University, ROOM 1-303, BLDG FIT, 100084 Beijing, China.

References

1. D. Reynolds, T.F. Quatieri, and R.B. Dunn, "Speaker verification using adapted gaussian mixture models," *Digital Signal Processing*, vol. 10, no. 1, pp. 19–41, 2000.
2. P. Kenny, G. Boulianne, P. Ouellet, and P. Dumouchel, "Joint factor analysis versus eigenchannels in speaker recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, pp. 1435–1447, 2007.
3. P. Kenny, G. Boulianne, P. Ouellet, and P. Dumouchel, "Speaker and session variability in gmm-based speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, pp. 1448–1460, 2007.
4. W.M. Campbell, D.E. Sturim, and D.A. Reynolds, "Support vector machines using gmm supervectors for speaker verification," *Signal Processing Letters, IEEE*, vol. 13, no. 5, pp. 308–311, 2006.
5. S. Ioffe, "Probabilistic linear discriminant analysis," *Computer Vision ECCV 2006, Springer Berlin Heidelberg*, pp. 531–542, 2006.
6. T. Kinnunen and H. Li, "An overview of text-independent speaker recognition: From features to supervectors," *Speech communication*, vol. 52, no. 1, pp. 12–40, 2010.
7. J. Li, D. Yu, J.T. Huang, and Y. Gong, "Improving wideband speech recognition using mixed-bandwidth training data in cd-dnn-hmm," *SLT*, pp. 131–136, 2012.
8. V. Ehsan, L. Xin, M. Erik, L. M. Ignacio, and G.-D. Javier, "Deep neural networks for small footprint text-dependent speaker verification," *IEEE International Conference on Acoustic, Speech and Signal Processing (ICASSP)*, vol. 28, no. 4, pp. 357–366, 2014.
9. P. Kenny, V. Gupta, T. Stafylakis, P. Ouellet, and J. Alam, "Deep neural networks for extracting baum-welch statistics for speaker recognition," *Odyssey*, 2014.
10. J. Wang, D. Wang, Z.-W. Zhu, T.F. Zheng, and F. Song, "Discriminative scoring for speaker recognition based on i-vectors," *APSIPA*, 2014.