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

Improving Deep Neural Networks Based Speaker Verification Using Unlabeled Data

Yao Tian[1], *Meng Cai*[2*], *Liang He*[1], *Wei-Qiang Zhang*[1], *Jia Liu*[1]

[1]National Laboratory for Information Science and Technology, Department of Electronic Engineering, Tsinghua University, Beijing, China
[2]Microsoft Research Asia, Beijing, China

tianyao11@mails.tsinghua.edu.cn, menca@microsoft.com
{heliang, zhangwq, liuj}@mail.tsinghua.edu.cn

Abstract

Recently, deep neural networks (DNNs) trained to predict senones have been incorporated into the conventional i-vector based speaker verification systems to provide soft frame alignments and show promising results. However, the data mismatch problem may degrade the performance since the DNN requires transcribed data (out-domain data) while the data sets (in-domain data) used for i-vector training and extraction are mostly untranscribed. In this paper, we try to address this problem by exploiting the unlabeled in-domain data during the training of the DNN, hoping the DNN can provide a more robust basis for the in-domain data. In this work, we first explore the impact of using in-domain data during the unsupervised DNN pre-training process. In addition, we decode the in-domain data using a hybrid DNN-HMM system to get its transcription, and then we retrain the DNN model with the "labeled" in-domain data. Experimental results on the NIST SRE 2008 and the NIST SRE 2010 databases demonstrate the effectiveness of the proposed methods.

Index Terms: deep neural networks, speaker verification, unlabeled data

1. Introduction

Over recent years, the Gaussian Mixture Model (GMM) has laid the foundation of many speaker verification systems [1, 2, 3], among which i-vector has become a dominant approach for speaker verification and has brought significant performance improvement. In i-vector paradigm, each utterance is represented as a low-dimensional vector called i-vector and then probabilistic linear discriminant analysis (PLDA) could be performed to get the final verification scores [4, 5, 6].

More recently, deep neural networks (DNNs) have become the state-of-the-art in automatic speech recognition (ASR) systems, bringing an about 30% relative improvement in word error rate (WER) [7, 8]. DNN approaches have also been evaluated in speaker verification area in the last few years and many of them try to use the DNN as a direct replacement for the classical GMM approach [9, 10, 11]. Recently, however, a hybrid framework proposed in [12] and [13] has shown promising results in speaker verification tasks where the DNN trained to discriminant between senones is used as a replacement of the GMM to provide frame posterior probabilities during the extraction of sufficient statistics. Their work actually demonstrate that more accurate content (senones) frame alignments could benefit speaker verification tasks. This framework has also been successfully applied in spoken language recognition [14].

Generally, speech recognition and speaker verification are two individual tasks and it is usually very difficult to have sufficient transcribed in-domain data that matches speaker verification task. The DNN model trained with the out-domain data only might not fully reflect the phonetic characteristics of the target acoustic space. Finding a way to effectively narrow the gap caused by data mismatch is thus a meaningful issue. In our previous work [15], we try to address this problem by using a GMM trained with bottleneck features of in-domain data as a replacement of the DNN trained with out-domain data to provide frame posterior probabilities. In this paper, however, we try to adapt the DNN model to the target acoustic space directly using the unlabeled (untranscribed) in-domain data. In our work, we firstly explore the effects of using in-domain data for DNN initialization during the unsupervised DNN pre-training process. In this way, we hope the region of space covered by the solution associated with this initialization could shrink to an area that is suitable both for out-domain data and in-domain data. In addition, we try to obtain the transcription of in-domain data automatically using a hybrid DNN-HMM system, and then retrain the DNN model with the "transcribed" in-domain data. Even though these machine-labeled transcriptions are not accurate enough, it can be inferred that the frames of in-domain data that have been grouped to the same senone are more phonetically correlated to each other and the DNN model trained with these data can provide more accurate content frame alignments to some extent during test stage. We also evaluate the performance of using bottleneck features extracted from our adapted DNNs for sufficient statistics extraction to see if additional improvement could be obtained. Experiments on the NIST SRE 2008 female short2-short3 English telephone task and the NIST SRE 2010 female core-extended English telephone task verify the effectiveness of the proposed methods.

The remainder of this paper is organized as follows. Section 2 presents the DNN based i-vector framework. Section 3 introduces our proposed strategies of using unlabeled in-domain data during the training of DNNs. Experimental setup and results are given in Section 4. Finally, conclusions are presented in Section 5.

2. The DNN based i-vector framework

The conventional i-vector model is based on the GMM-UBM and each utterance is represented by its sufficient statistics

*This work was conducted during his doctorate study period in Tsinghua University.

http://dx.doi.org/10.21437/Interspeech.2016-614

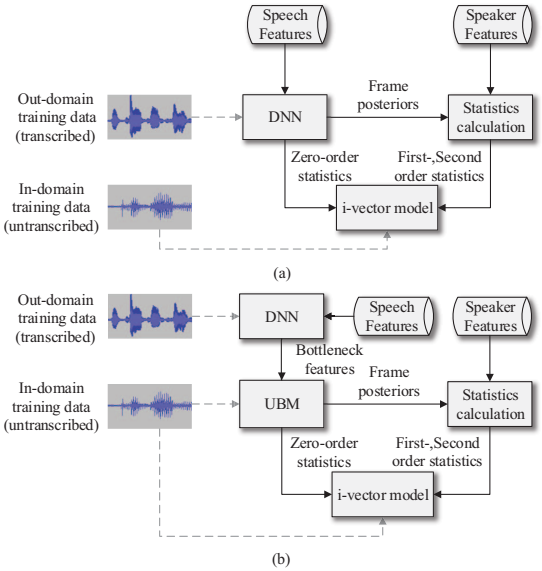

Figure 1: *The flow diagrams of (a) DNN based (b) bottleneck features based i-vector framework.*

(Baum-Welch statistics) extracted with the UBM as follows

$$N_c = \sum_t \gamma_{c,t} \quad (1)$$

$$\mathbf{F}_c = \sum_t \gamma_{c,t}\mathbf{o}_t \quad (2)$$

$$\mathbf{S}_c = \sum_t \gamma_{c,t}\mathbf{o}_t\mathbf{o}_t^T \quad (3)$$

where N_c, $\mathbf{F}_c$ and $\mathbf{S}_c$ are the zero-order, first-order and second statistics of an utterance corresponding to the c-th Gaussian component of UBM. $\gamma_{c,t}$ is the posterior probability of feature vector $\mathbf{o}_t$ generated by the c-th Gaussian component.

In [12] and [13], the DNN trained to predict senones is used as a replacement of the GMM-UBM in extracting sufficient statistics. Specifically, by treating each senone outputs of the DNN as a single-Gaussian like units in the UBM, the sufficient statistics could be extracted by replacing the Gaussian posterior with senone posterior. This could be regarded as phonetically-dependent (senone-dependent) frame alignments where content based comparison is able to be made between different speakers afterwards.

3. The bottleneck features based i-vector framework

The DNN based bottleneck (BN) features are being widely used in various speech related applications to improve system's performance [16, 17, 18]. In our previous work, we propose using a GMM trained with BN features to provide frame alignments to address the data mismatch problem [15]. Specifically, the senone-based DNN trained with out-domain data is used as a BN feature extractor. Then a GMM is trained in the traditional unsupervised way with BN features of in-domain data and is used to calculate frame posterior probabilities while collecting sufficient statistics. The BN feature contains rich phonetic information since the DNN is trained to discriminant between senones. As a result, each component of the GMM is more connected to phonetic content. In addition, the GMM alignment model can be more accurate than the DNN alignment model in modeling the target acoustic space to some extent by utilizing the in-domain data. The flow diagrams of DNN based and BN features based i-vector framework are presented in Figure 1.

4. Utilizing unlabeled in-domain data

The BN features based approach can be viewed as an indirect model adaptation of the higher layers of the DNN. In this paper, we try to adapt the whole DNN model to the target acoustic space directly by exploiting unlabeled in-domain data.

4.1. unsupervised pre-training with in-domain data

The training of DNN is a two stage process: unsupervised pre-training and supervised fine-tuning [19]. The pre-training can make a very important difference because of the non-convexity of the training criterion. The idea of pre-training the DNN is to regard it as a generative Deep Belief Network (DBN) which is a stack of Restricted Boltzman Machines (RBMs) [20]. Since the pre-training plays an important role and is implemented in an unsupervised way, it is interesting to investigate the effects of reinforcing pre-training by utilizing unlabeled in-domain data. The pre-training actually can be viewed as an unsupervised learning process of feature representation. More abstract feature representation could be learned as the neural network becomes deeper. By utilizing the unlabeled in-domain data together with out-domain data is expected to make the neural network a more robust feature detector. On the other hand, back-propagation algorithm used during fine-tuning step is actually more effective for the higher layer parameters and less effective for the lower layer parameters. The fine-tuning may be more effective for both in-domain and out-domain data if the parameters of lower layer are robustly learned. In other words, the DNN may therefore provide more accurate frame alignments for in-domain data.

4.2. supervised re-training with in-domain data

The current state-of-the-art DNN based ASR systems usually require a relatively large transcribed training data to be trained on to make full use of DNN's discriminative ability. However, the data preparation costs too much time and human efforts, which can be prohibitive especially for languages with few speakers. A practical way in ASR to address this problem is using self re-training methods [21, 22, 23]. In these methods, the transcribed data are firstly used to build a seed model. Then this model is used to decode the untranscribed data to generate transcriptions, which are regarded as ground-truth transcripts in further training.

In this paper, we investigate using the self re-training strategy to address the data mismatch problem. In our work, we first train a hybrid DNN-HMM system using the transcribed out-domain data to decode the untranscribed in-domain data. Then we retrain the DNN model with the "transcribed" in-domain data to provide frame posteriors. Even though the senone transcripts of the in-domain data generated by ASR system may not be accurate enough, it can be inferred that the frames of in-domain data that have been grouped to the same senone are more phonetically correlated to each other. As a result, it can be expected that the DNN model trained with these data might be more effective in reflecting the phonetic characteristics of the target acoustic space, thus providing more accurate content frame alignments to some extent.

5. Experiments

5.1. Experimental setup

5.1.1. Dataset

Experiments are carried out on the NIST SRE2008 female short2-short3 telephone-telephone English task (Condition 7) and the NIST SRE2010 female core-extended telephone-telephone English task (Condition 5) [24, 25]. The training data for senone-based DNNs are 300 hours English telephone speeches from Switchboard-I (out-domain data). The training data for DNN unsupervised pre-training, DNN supervised re-training, UBM, T matrix and PLDA are selected from NIST SRE 04, 05, 06 telephone data (in-domain data).

5.1.2. Models

- **GMM-HMM**: A GMM-HMM system is firstly trained to generate transcriptions for senones. It uses 52-dimensional PLP features (13 basic + first-order + second-order + third-order) with speaker-based mean-covariance normalization. Then the features are further reduced to 39 dimension by HLDA. The GMM-HMM uses 2227 senones tied by a phonetic decision tree.
- **DNN**: The DNN used to provide the posterior probability is pre-trained as Deep Belief Network (DBN) and then is fine-tuned with cross-entropy criterion. 11 frames where each frame consists of 120 log Mel-filterbank coefficients (40 basic + first-order + second order) are concatenated as the input of the network. The DNN has five hidden layers and each hidden layer has 1200 nodes. The output of the DNN with respect to senones has 2227 nodes. The configurations of the DNN with BN layer are the same except that the number of the fifth hidden layer is changed to 39.
- **DNN-HMM**: The DNN-HMM system with 2227 senones is used for the decoding of the unlabeled in-domain data. A trigram language model is trained using the transcriptions of the 2000-hour English Fisher corpus with modified Knerser-Ney smoothing and is interpolated with a more general trigram.
- **UBM-i-vector model**: 39-dimensional (13 basic + first order + second order) PLP is extracted as the raw acoustic feature. Then a gender-dependent diagonal covariance UBM with 2048 mixtures is trained. The dimensionality of i-vectors is 400. Simplified Gaussian PLDA is used to generate verification scores and the dimensionality of speaker subspace in PLDA model is 200.
- **DNN-ivector model**: The DNN with senone outputs are used to provide frame posteriors. Then these frame posteriors are combined with 39-dimensional PLP features for sufficient statistics extraction. The number of mixtures is confined by the number of senones. Other model configurations are the same with the UBM-ivector model.
- **BN-ivector model**: the DNN with senone outputs are used to extract 39-dimensional BN features. Then a UBM is trained with these BN features and is used to provide frame posteriors. These frame posteriors are combined with 39-dimensional PLP features for sufficient statistics extraction. The number of mixtures is set to 2048 for all systems. Other model configurations are the same with the UBM-ivector model.

Equal error rate (EER) and minimum decision cost function (minDCF) are adopted for evaluation [24].

5.2. Experimental results

Table 1: *Results of UBM-i-vector and DNN-i-vector with different DNN pre-training data on the NIST SRE08 short2-short3 condition7 and the NIST SRE10 core-extended condition5, in terms of EER(%)/minDCF$times$10. The last column gives the word error rate (WER(%)) of corresponding DNN-HMM systems on Hub5'00-SWB.*

system	SRE08	SRE10	ASR
UBM-i-vector	2.02/0.106	3.12/0.156	
DNN_{SWB}-i-vector	1.81/0.089	2.84/0.141	18.8
DNN_{SRE}-i-vector	1.90/0.088	3.05/0.147	19.0
$DNN_{SRE+SWB}$-i-vector	1.72/0.084	2.71/0.138	18.8

We first evaluate the effects of using different data for DNN pre-training. Table 1 presents the results of UBM-i-vector and DNN-ivector with different pre-training data. The SRE data used for pre-training are 300 hours randomly selected speeches from NIST SRE 04, 05, 06 and all DNN models are fine-tuned with SWB data. It can be seen that all the DNN based systems outperform the UBM approach. However, the DNN_{SRE} system is slightly worse compared with the other two DNN systems since the data for pre-training and fine-tuning are completely different and the fine-tuning of the DNN might be less effective. In addition, the $DNN_{SWB+SRE}$ based system can provide further performance improvements over the DNN_{SWB} system, and the relative improvements are 4.97% in EER, 5.62% in minDCF on SRE08 trial, 4.58% in EER, 2.13% in minDCF on SRE10 trial. We also evaluate the performance of different DNN based ASR systems on Hub5'00-SWB speech recognition data set. The results are presented in Table 1 as well. The performance of $DNN_{SWB+SRE}$ and DNN_{SWB} based ASR systems are the same while the DNN_{SRE} is slightly worse. From the above speech recognition and speaker verification results, it can be concluded that DNN pre-trained with both in-domain data and out-domain data is robust and effective for both tasks.

Table 2: *Results of UBM-i-vector, DNN-i-vector and BN-ivector with different DNN pre-training data on the NIST SRE08 short2-short3 condition7 and the NIST SRE10 core-extended condition5, in terms of EER(%)/minDCF08$\times$10.*

system	SRE08	SRE10
UBM-i-vector	2.02/0.106	3.12/0.156
DNN_{SWB}-i-vector	1.81/0.089	2.84/0.141
BN_{SWB}-i-vector	1.69/0.083	2.70/0.136
$DNN_{SRE+SWB}$-i-vector	1.72/0.084	2.71/0.138
$BN_{SRE+SWB}$-i-vector	1.58/0.079	2.62/0.132

Table 2 presents the results of UBM-i-vector, DNN-ivector and BN-i-vector with different pre-training data. From the results we can see the $DNN_{SRE+SWB}$-i-vector is competitive to BN_{SWB}-i-vector. Additional performance improvements could be obtained with BN features extracted from $DNN_{SRE+SWB}$, and the relative improvements are 8.14% in EER, 5.95% in minDCF on SRE08 trial, 3.32% in EER, 4.35% in minDCF on SRE10 trial. Since the bottleneck layer in our DNN is the last hidden layer, the pre-training can be viewed as an adaptation of parameters below the bottleneck layer and the BN based approach can be viewed as an adaptation of parameters above

the bottleneck layer. Thus combining these two complementary methods will lead to more accurate frame alignments for the in-domain data, which explains the additional performance improvement of $BN_{SRE+SWB}$-i-vector over $DNN_{SRE+SWB}$-i-vector.

Table 3: *Results of UBM-i-vector and DNN-i-vector with different DNN fine-tuning data on the NIST SRE08 short2-short3 condition7 and the NIST SRE10 core-extended condition5, in terms of EER(%)/minDCF08×10. The last column gives the Word Error Rate (WER(%)) of corresponding DNN-HMM systems on Hub5'00-SWB.*

system	SRE08	SRE10	ASR
UBM-i-vector	2.02/0.106	3.12/0.156	
DNN_{SWB}-i-vector	1.72/0.084	2.71/0.138	18.8
DNN_{SRE}-i-vector	1.47/0.076	2.43/0.125	29.3
$DNN_{SRE+SWB}$-i-vector	1.64/0.079	2.50/0.129	22.1

Next, we evaluate the effects of using "transcribed" in-domain data for DNN fine-tuning. Table 3 presents the results of DNN-ivector with different fine-tuning data. The SRE data that have been decoded for fine-tuning are the same as the data used in unsupervised pre-training experiments. The DNN-HMM system used for decoding the SRE data is trained with SWB data. The DNNs in Table 3 are all pre-trained with both SWB and SRE data. From the results we can see that DNN_{SRE} based system performs much better than DNN_{SWB} based approach, the relative improvements are 14.5% in EER, 9.52% in minDCF on SRE08 trial, 10.3% in EER, 9.42% in minDCF on SRE10 trial. However, the performance of $DNN_{SRE+SWB}$-i-vector is not as good as DNN_{SRE}-i-vector though it outperforms DNN_{SWB}-i-vector. From the results it can be concluded that DNN fine-tuned with SRE data do reflect the phonetic characteristics of the target acoustic space better than DNN fine-tuned with SWB data. In addition, the results of different DNN based ASR systems on Hub5'00-SWB speech recognition data set are presented in Table 3. From the results we can see that for ASR task, the DNN based systems perform much worse when the SRE data are included for fine-tuning. Actually, it can be inferred that the frames of SRE data that have been grouped to the same senone are more phonetically correlated to each other, but they may not strictly match the senone labels that have been assigned to them. As a result, the DNN trained with these "transcribed" SRE data are more relevant and reliable for speaker verification task while the inclusion of them will be harmful to speech recognition task.

Table 4: *Results of DNN-i-vector and BN-ivector with different DNN fine-tuning data on the NIST SRE08 short2-short3 condition7 and the NIST SRE10 core-extended condition5, in terms of EER(%)/minDCF08×10.*

system	SRE08	SRE10
UBM-i-vector	2.02/0.106	3.12/0.156
DNN_{SWB}-i-vector	1.72/0.084	2.71/0.138
BN_{SWB}-i-vector	1.58/0.079	2.62/0.132
DNN_{SRE}-i-vector	1.47/0.076	2.43/0.125
BN_{SRE}-i-vector	1.88/0.089	2.90/0.143
$DNN_{SRE+SWB}$-i-vector	1.64/0.079	2.50/0.129
$BN_{SRE+SWB}$-i-vector	1.56/0.077	2.52/0.128

Table 4 presents the results of DNN-ivector and BN-i-vector with different fine-tuning data. The BN feature based systems outperform the DNN based approaches when SWB data are included for DNN fine-tuning. However, the BN features based system performs much worse when only SRE data are used for DNN fine-tuning. A reasonable explanation might be that BN features extracted from SRE data related DNNs are less phonetically discriminative compared with BN features extracted from DNN trained with SWB data since the transcripts of SRE data are not accurate enough. As a result, the GMM trained with these BN features might be less effective for acoustic space modeling. Besides, since the difference between DNN_{SRE} and BN_{SRE} based approaches is that the discriminative output (trained with supervision) has been replaced by unsupervised clustering of BN features in a GMM, it can be inferred that the softmax layer of DNN_{SRE} plays important role for frame alignments and contains much phonetic information with respect to the target acoustic space.

Table 5: *Results of DNN-ivector fine-tuned with randomly selected SRE data and sentence-level confidence based SRE data on the NIST SRE08 short2-short3 condition7 and the NIST SRE10 core-extended condition5, in terms of EER(%)/minDCF08×10.*

system	SRE08	SRE10
UBM-i-vector	2.02/0.106	3.12/0.156
$DNN_{SRE\text{-}random}$-i-vector	1.47/0.076	2.43/0.125
$DNN_{SRE\text{-}confidence}$-i-vector	1.42/0.073	2.35/0.120

Generally, it is necessary to take confidence measure into consideration when choosing in-domain data for model re-training. We first decode all the SRE training data and then select 300 hours speeches with highest sentence confidence for model re-training for comparison. The sentence confidence is calculated as the average word confidence in a sentence. The results are presented in Table 5. It can be seen that data selection could bring additional performance improvements. The $DNN_{SRE\text{-}confidence}$-i-vector performs best among all systems evaluated in this paper.

6. Conclusions

In this paper we try to address the data mismatch problem that may arise in the hybrid DNN-ivector framework by using unlabeled in-domain data. Experiments on the NIST SRE 2008 and 2010 female English telephone tasks show that DNN pre-trained with both in-domain and out-domain data is more effective for speaker verification. Using BN features extracted from this DNN to calculate frame posteriors can provide further improvements. In addition, substantial performance improvements can be obtained when the DNN fine-tuned with decoded in-domain data is used for frame alignments. However, the BN feature extracted from DNNs in this situation is less effective. In the future, we'll analyze the effects of using different number of decoded data and evaluate systems' performance on multilingual speaker verification tasks.

7. Acknowledgements

This work was supported by the National Natural Science Foundation of China under Grant No. 61273268, No. 61370034 and No. 61403224.

8. References

[1] 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, no. 4, pp. 1435–1447, 2007.

[2] 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.

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

[4] S. J. Prince and J. H. Elder, "Probabilistic linear discriminant analysis for inferences about identity," in *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*. IEEE, 2007, pp. 1–8.

[5] P. Kenny, "Bayesian speaker verification with heavy-tailed priors." in *Odyssey*, 2010, p. 14.

[6] D. Garcia-Romero and C. Y. Espy-Wilson, "Analysis of i-vector length normalization in speaker recognition systems." in *Interspeech*, 2011, pp. 249–252.

[7] G. E. Dahl, D. Yu, L. Deng, and A. Acero, "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 20, no. 1, pp. 30–42, 2012.

[8] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *Signal Processing Magazine, IEEE*, vol. 29, no. 6, pp. 82–97, 2012.

[9] T. Stafylakis, P. Kenny, M. Senoussaoui, and P. Dumouchel, "Preliminary investigation of boltzmann machine classifiers for speaker recognition." in *Odyssey*, 2012, pp. 109–116.

[10] M. Senoussaoui, N. Dehak, P. Kenny, R. Dehak, and P. Dumouchel, "First attempt of boltzmann machines for speaker verification." in *Odyssey*, 2012, pp. 117–121.

[11] S. Garimella and H. Hermansky, "Factor analysis of auto-associative neural networks with application in speaker verification," *Neural Networks and Learning Systems, IEEE Transactions on*, vol. 24, no. 4, pp. 522–528, 2013.

[12] Y. Lei, N. Scheffer, L. Ferrer, and M. McLaren, "A novel scheme for speaker recognition using a phoneetically-aware deep neural network," *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, 2014.

[13] P. Kenny, T. Stafylakis, P. Ouellet, V. Gupta, and J. Alam, "Deep neural networks for extracting Baum-Welch statistics for speaker recognition," in *Odyssey 2014*, 2014, pp. 293–298.

[14] L. Ferrer, Y. Lei, M. McLaren, and N. Scheffer, "Study of senone-based deep neural network approaches for spoken language recognition," *Audio, Speech, and Language Processing, IEEE/ACM Transactions on*, vol. 24, no. 1, pp. 105–116, 2016.

[15] Y. Tian, M. Cai, L. He, and J. Liu, "Investigation of bottleneck features and multilingual deep neural networks for speaker verification." in *Interspeech*, 2015, pp. 1151–1155.

[16] D. Yu and M. L. Seltzer, "Improved bottleneck features using pretrained deep neural networks." in *Interspeech*, 2011, p. 240.

[17] Y. Song, B. Jiang, Y. Bao, S. Wei, and L.-R. Dai, "I-vector representation based on bottleneck features for language identification," *Electronics Letters*, vol. 49, no. 24, pp. 1569–1570, 2013.

[18] J. Gehring, Y. Miao, F. Metze, and A. Waibel, "Extracting deep bottleneck features using stacked auto-encoders," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 3377–3381.

[19] Y. Bengio, "Learning deep architectures for ai," *Foundations and trends® in Machine Learning*, vol. 2, no. 1, pp. 1–127, 2009.

[20] G. Hinton, "A practical guide to training restricted boltzmann machines," *Momentum*, vol. 9, no. 1, p. 926, 2010.

[21] S. Novotney, R. Schwartz, and J. Ma, "Unsupervised acoustic and language model training with small amounts of labelled data," in *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*. IEEE, 2009, pp. 4297–4300.

[22] Y. Huang, D. Yu, Y. Gong, and C. Liu, "Semi-supervised gmm and dnn acoustic model training with multi-system combination and confidence re-calibration." in *INTERSPEECH*, 2013, pp. 2360–2364.

[23] K. Vesely, M. Hannemann, and L. Burget, "Semi-supervised training of deep neural networks," in *Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on*. IEEE, 2013, pp. 267–272.

[24] "The NIST year 2008 speaker recognition evaluation plan," *http://www.itl.nist.gov/iad/mig//tests/sre/2008/*, 2008.

[25] "The NIST year 2010 speaker recognition evaluation plan," *http://www.nist.gov/speech/tests/spk/2010/index.html*, 2010.