A Scalable Approach to Using DNN-Derived Features in GMM-HMM Based Acoustic Modeling For LVCSR

Zhi-Jie Yan *Qiang Huo* *Jian Xu**

Microsoft Research Asia, Beijing, China

{zhijiey,qianghuo,v-jiaxu}@microsoft.com

Abstract

We present a new scalable approach to using deep neural network (DNN) derived features in Gaussian mixture density hidden Markov model (GMM-HMM) based acoustic modeling for large vocabulary continuous speech recognition (LVCSR). The DNN-based feature extractor is trained from a subset of training data to mitigate the scalability issue of DNN training, while GMM-HMMs are trained by using state-of-the-art scalable training methods and tools to leverage the whole training set. In a benchmark evaluation, we used 309-hour Switchboard-I (SWB) training data to train a DNN first, which achieves a word error rate (WER) of 15.4% on NIST-2000 Hub5 evaluation set by a traditional DNN-HMM based approach. When the same DNN is used as a feature extractor and 2,000-hour "SWB+Fisher" training data is used to train the GMM-HMMs, our DNN-GMM-HMM approach achieves a WER of 13.8%. If per-conversation-side based unsupervised adaptation is performed, a WER of 13.1% can be achieved.

Index Terms: deep neural network, DNN-based feature extraction, DNN-GMM-HMM, DNN-HMM, LVCSR

1. Introduction

In the past several years, a so-called DNN-HMM approach has become a new state-of-the-art acoustic modeling method for large vocabulary continuous speech recognition (LVCSR) (e.g., [1–5]). The main factors contributed to the improved recognition accuracy compared with the traditional GMM-HMM based approach include the use of long-span features in the input layer of a DNN, a hierarchical highly nonlinear feature mapping due to its deep structure, and using decision-tree based HMM tied-states as target classes in DNN output layer. Our recent study in [6] has shown that when long-span features and a tied-state based discriminative training criterion are used, it is possible for the GMM-HMM approach to achieve similar state classification accuracy on training set as that of the DNN-HMM approach. However, there is still a performance gap in terms of word error rate (WER) between DNN-HMM and GMM-HMM approaches on the testing set. This may suggest that the hierarchical nonlinear feature extraction capability of a DNN is the most important contributing factor in the success of DNN-HMM approach.

In literature, there are several methods which derive features from either a shallow or a deep neural network, and use the features in GMM-HMM based acoustic modeling. For example, a so-called tandem approach [7] uses log-posterior probabilities generated by the softmax output layer of a multi-layer perceptron (MLP), or the weighted sums of the outputs of the last hidden layer as features. A so-called bottleneck-feature approach [8] imposes a bottleneck in the middle of an MLP, and uses the weighted sums of the outputs of the hidden layer immediately before the bottleneck layer as features. More recently, encouraged by the success of DNN-HMM approach, several extensions to the conventional tandem and bottleneck-feature approaches are proposed, which include using a bottleneck in a pre-trained DNN in [9], and using auto-encoder bottleneck-features on top of a DNN in [10]. However, to the best of our knowledge, none of them has outperformed the best DNN-HMM solution yet in a comparable setup [9–11].

Given the preliminary success of DNN-HMM approach, one of the most important research problems is how to scale up DNN training to leverage big data and further improve recognition accuracy. Past and ongoing efforts include using multiple GPU cards (e.g., [12]) and large-scale CPU clusters (e.g., [13]) to speed up DNN training, and exploring new scalable optimization methods for DNN training (e.g., [4, 13]).

In this paper, we present a new scalable approach to using DNN-derived features in GMM-HMM based acoustic modeling for LVCSR. The DNN is used as a feature extractor which is trained from a sampled subset of training data to mitigate the scalability issue of DNN training, while GMM-HMMs are trained by using state-of-the-art scalable training methods to make use of the whole training set. In a benchmark evaluation on Switchboard-I conversational telephone speech transcription task, it is demonstrated for the first time that our DNN-GMM-HMM approach can outperform the traditional DNN-HMM approach and offers a promising practical solution.

The rest of this paper is organized as follows. In Section 2, we present the details of our new approach. In Section 3, we report our experimental results. Finally, we conclude the paper in Section 4.

2. Our Approach

2.1. Training a DNN for feature extraction

The topology of a DNN is defined by the number of hidden layers and the number of nodes in its input, hidden, and output layers, respectively. Typically, the number of nodes in the input layer is determined by the length of input feature vector, which is composed by augmenting several neighboring frames of the original D_{ori}-dimensional feature vectors. The number of hidden layers and the number of nodes in each hidden layer are set empirically by considering the tradeoff among trainability, classification accuracy and runtime efficiency. In our approach, the number of nodes in the output layer is set according to an initial GMM-HMM set trained on the same task. Using traditional training procedures, the GMM-HMM set is trained on the full training set using spectral features. The GMM-HMM

*Jian Xu contributed to this paper when he worked as an intern in the Speech Group of Microsoft Research Asia.

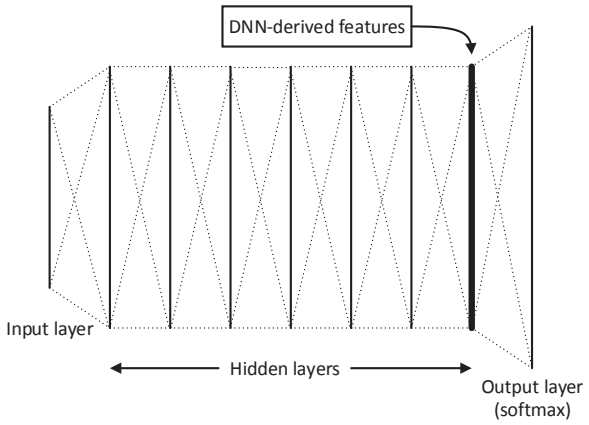

Figure 1: *Illustration of a typical 9-layer (7-hidden) DNN and the layer from which the features are derived.*

topology and the structure of decision-tree based state-tying are well-tuned empirically. The number of nodes in the DNN output layer is then equal to the number of tied-states in the corresponding GMM-HMM set.

To train the DNN feature extractor, a well-established DNN training recipe using cross entropy (CE) criterion can be used (e.g., [1, 2]). The initial feature-state alignment required for DNN training is generated by performing forced-alignment using spectral features and the initial GMM-HMM set. Only a sampled subset of training data is used in this step to mitigate the scalability issue of DNN training. Different sampling strategies can be used to obtain the subset, for example, random utterance- or frame-level sampling, tied-state distribution preserving sampling, WER-guided sampling, among other possibilities. Such trained DNN can be used directly to perform speech recognition as in DNN-HMM approach (e.g., [1, 2]), or it can be used to extract features for GMM-HMM acoustic modeling as described in the following subsections.

2.2. Deriving features from a DNN

Fig. 1 illustrates how features are derived from a trained DNN in our approach. Firstly, a D_{raw}-dimensional raw feature vector is obtained from the last hidden layer by calculating the weighted sums of the outputs of its previous layer. In this way, a new set of DNN-derived features can be extracted for the full training set. After that, principal component analysis (PCA) is performed to compress the dimension of the DNN-derived features to D_{dnn}, and each feature frame is augmented with its original spectral features to compose a new $(D_{\text{dnn}} + D_{\text{ori}})$-dimensional feature vector. Finally, HLDA [14] is performed to reduce the feature dimension to D_{gmm}, which is appropriate for GMM-HMM acoustic modeling. In this step, the same set of decision-tree based tied-states in the initial GMM-HMM set is used to define the "classes" in training HLDA transform.

2.3. GMM-HMM training using DNN-derived features

The GMM-HMM training using DNN-derived features is quite straightforward. The same decision-tree state-tying structure is used again, and a single-pass re-estimation is performed using both spectral and DNN-derived features in parallel, to initialize a single-Gaussian GMM for each HMM tied-state. Conventional maximum likelihood (ML) based GMM-HMM training recipe is then applied to grow the number of mixture components of the GMMs to an appropriate value.

After ML training, discriminative training is performed in two steps. Firstly, we follow our previous work in [6] to perform a lattice-free, tied-state based WE-RDLT training in the feature space. A set of contextual weight expanded linear transforms is estimated using a maximum mutual information (MMI) criterion, which is identical to the cross entropy criterion in DNN training. Secondly, conventional sequence-based MMI training is performed in the model space (e.g., [15]). Language model scores encoded in word lattices are incorporated in this step to refine the GMM-HMM parameters.

In certain application scenarios, unsupervised adaptation can be performed in the recognition stage. Therefore, a second-pass decoding is conducted to further improve the recognition accuracy. In this study, a traditional CMLLR adaptation approach [16] is used to demonstrate the impact of unsupervised adaptation.

2.4. Discussion

Our method of deriving features from a DNN is different from the conventional tandem or bottleneck-feature approaches. Because the bottleneck layer is no longer necessary in our approach, the power of nonlinear discriminative feature extraction in DNN can be fully utilized. In our approach, the dimension of the raw DNN-derived feature vector is independent of the number of nodes in the output layer, which is not the case for the tandem approach. Therefore, the trained DNN feature extractor can be used in other tasks or setups with a different decision-tree tying structure. Given the promising results in [17], we conjecture that it is possible to train a DNN feature extractor which can be shared by GMM-HMM based LVCSR systems for different languages. Given the experimental evidence in [18], it is also possible to train a DNN feature extractor by using both clean and noisy speech data to improve the noise robustness of a GMM-HMM based LVCSR system.

In addition to mitigating the scalability issue, using GMM-HMM to model DNN-derived features provides additional opportunities of further improving recognition accuracy by leveraging some well-established techniques in GMM-HMM framework such as adaptation (e.g., [19] and the references therein), adaptive training (e.g., [16, 20]) or irrelevant variability normalization (IVN) based training (e.g., [21–23]), discriminative feature extraction (e.g., [6, 24, 25]), among others.

3. Experiments

3.1. Experimental setup

Two training data sets are used in this study. The first is a 309-hour set from Switchboard-I conversational telephone speech transcription task [26]. The second is a 2,000-hour set which is composed of the aforementioned 309-hour set and another 1,700 hours speech from Fisher English corpus (parts 1 and 2) [27]. The 1831-segment SWB part of the NIST 2000 Hub5 evaluation set which consists of 40 conversational sides (about 2 hours of speech) is used as the testing set.

For front-end spectral feature extraction, we use 13-dimensional PLP features along with their time derivatives up to the third order, i.e., $D_{\text{ori}} = 13 \times 4 = 52$. Windowed mean and variance normalization is performed, and a 39×52 HLDA transform is estimated to reduce the feature dimension for GMM-HMM modeling, i.e., $D_{\text{gmm}} = 39$.

Using spectral features, two sets of speaker independent GMM-HMMs are trained. The first is trained using the 309-

Table 1: *WER (%) comparison of different systems using 309-hour training data (UA = unsupervised adaptation).*

DNN-HMM (9x2k, 9.3k, 309hr)	DNN-GMM-HMM (9.3k, 309hr)			
	ML	RDLT	MMI	UA
16.4	17.8	16.1	15.3	14.7

hour data set, and has about 9.3k phonetic decision-tree based tied triphone HMM states, each modeled by a 40-component GMM. The second is trained using the 2,000-hour data set, and has about 18k tied-states, each modeled by a 72-component GMM.

For DNN training, 11 augmented frames of the original 52-dimensional spectral features are fed into the input layer. The training procedure is the same as in [2]. Several DNNs with different topologies are trained to investigate the effect of several structural parameters.

For the GMM-HMM sets trained using DNN-derived features, the topology and other setups are kept the same as that of using spectral features. The WE-RDLT [6] training is performed using 1,000 "regions." Unsupervised adaptation is done for each conversation side, where CMLLR with 2 regression classes, one for speech and another for non-speech, is used.

It is a common practice to re-balance acoustic and language model scores when using features generated by neural networks [8]. In our experiments, an acoustic down-scaling factor of 0.5 is used in decoding and word-lattice based sequence training (simply multiply the acoustic log-likelihoods by 0.5). This is found essential to obtain the best recognition accuracy.

Our recognition vocabulary contains 47,633 unique words. The pronunciation lexicon contains multiple pronunciations per word with a total of 58,393 unique pronunciations. A trigram language model, which is trained on the 2,000-hour data transcripts and interpolated with a written-text trigram, is used in decoding. All of the recognition experiments are performed with a Microsoft in-house decoder and the results are evaluated by using the NIST Scoring Toolkit SCTK [28].

3.2. Experimental results

3.2.1. 309-hour experiments

We start from a large 9-layer (7-hidden) DNN which is trained using cross entropy criterion and 309-hour data set. The DNN has 2,048 nodes in each hidden layer and about 9.3k nodes in the output layer. We label this DNN as "9x2k, 9.3k, 309hr" and similarly for other DNN topologies hereinafter. Using the features derived from it, a GMM-HMM set which has the same tied-states (about 9.3k) is trained, where each state is modeled by a 40-component GMM.

The recognition performance (WER in %) of different systems is compared in Table 1. When used directly in the hybrid mode, the DNN-HMM achieves a WER of 16.4%, which is significantly better than the corresponding initial GMM-HMM systems trained using spectral features only (which achieves a WER of 26.1% after ML training and 20.5% after discriminative training). Using the features derived from this DNN, the WER of the ML-trained DNN-GMM-HMM system is improved significantly to 17.8%, but still worse than the DNN-HMM system. After WE-RDLT, the WER of our DNN-GMM-HMM system is reduced to 16.1%, which means the GMM-HMM using DNN-derived features can perform slightly better than its DNN-HMM counterpart. At this point, both the RDLT and DNN are trained with a tied-state based cross entropy criterion, and the GMM parameters are still ML-trained. By performing sequence MMI training to optimize the GMM parameters, the WER of our DNN-GMM-HMM system further reduces to 15.3%. Finally, after unsupervised CMLLR adaptation, the WER of the DNN-GMM-HMM system is reduced to 14.7%. The overall relative performance improvement is quite similar to the gain commonly obtained when performing discriminative training and adaptation on GMM-HMMs trained with spectral features.

Table 2: *WER (%) comparison of systems with different DNN topologies on 309-hour task. About 9.3k tied states are used in both DNN-HMM and DNN-GMM-HMM systems.*

DNN-HMM		DNN-GMM-HMM	
9x2k	16.4	ML Training	17.8
7x2k	16.7		18.0
7x1k	17.8		18.9
5x2k	17.9		21.5

Table 3: *WER (%) comparison of different systems with bottleneck-features and proposed DNN-derived features on 309-hour task. About 9.3k tied states are used in both DNN-HMM and DNN-GMM-HMM systems (BN = bottleneck).*

DNN-HMM		DNN-GMM-HMM	
7x2k	16.7	ML Training	18.0
7x2k-BN	18.0		19.6

3.2.2. Effect of number of hidden layers and hidden nodes

To investigate how different DNN topologies would affect the DNN-HMM and DNN-GMM-HMM systems, the numbers of hidden layers and hidden nodes are varied in our experiments. The recognition performance of different systems is compared in Table 2. When the numbers of hidden layers and hidden nodes of the DNN are reduced, the recognition performance of the DNN-HMM systems degrades gradually. The performance of the ML trained DNN-GMM-HMM systems follows similar trend. The experimental results demonstrate that the DNN benefits most from its highly nonlinear discriminative feature mapping brought by its deep structure. Wider hidden layers with more nodes also contribute to the performance. On this task, the 7x2k DNN feature extractor seems to be a good choice because it saves 1/3 of the runtime feature extraction cost compared with the 9x2k DNN (4 *vs.* 6 nonlinear layers), yet it achieves similar recognition performance after GMM-HMM modeling (17.8% *vs.* 18.0%). In real applications, tradeoff between efficiency and accuracy can be made to determine the appropriate DNN topology for feature extraction.

3.2.3. Comparison with deep bottleneck-features

In order to compare our DNN-derived features with the deep bottleneck-features in [9], a symmetric 7-layer (5-hidden) DNN which has a 52-node bottleneck in its 4th layer is trained. When used as a feature extractor, the raw bottleneck-features after PCA are augmented with the original spectral features. HLDA is then used to perform dimension reduction similarly as in our proposed approach. The comparison of recognition performance is given in Table 3. It can be seen as expected that

Table 4: *WER (%) comparison of different systems using 309-hour data for DNN training and 2,000-hour data for GMM-HMM training. About 18k tied states are used in both DNN-HMM and DNN-GMM-HMM systems (UA = unsupervised adaptation).*

DNN-HMM (9x2k, 18k, 309hr)	DNN-GMM-HMM (18k, 2000hr)			
	ML	RDLT	MMI	UA
15.4	16.1	14.7	13.8	13.1

Table 5: *WER (%) comparison of different systems using 2,000-hour data for training both DNN and GMM-HMMs. About 18k tied states are used in both DNN-HMM and DNN-GMM-HMM systems (UA = unsupervised adaptation).*

DNN-HMM (9x2k, 18k, 2000hr)		DNN-GMM-HMM (18k, 2000hr)			
CE	MMI	ML	RDLT	MMI	UA
14.6	13.3	15.6	14.5	13.0	12.3

the bottleneck layer hurts the performance of the DNN-HMM system by 1.3% absolute (or 7.8% relative). Even larger performance degradation (1.6% absolute, or 8.9% relative) is observed when the DNN is used for feature extraction. The results suggest that deriving features from a deeper hierarchy of hidden layers is better than from a bottleneck layer in the middle. The information loss caused by the bottleneck layer is detrimental to the performance, therefore should be avoided.

3.2.4. 2,000-hour experiments

Although the DNN-GMM-HMM system using DNN-derived features is able to achieve similar or even better performance than the DNN-HMM in the previous 309-hour experiments, the practical value of this combination is still limited. Using GMM-HMM to model DNN-derived features introduces additional procedures in training, while the computational cost in runtime feature extraction and decoding is not reduced either. Therefore, it is much more interesting to scale up the data used in GMM-HMM training and see how it performs using the combination suggested in this paper.

Using spectral features, the 2,000-hour full data set is used in our experiments to train a GMM-HMM set with about 18k tied states and 72 Gaussian mixture components per state. After that, a 9x2k (7-hidden) DNN feature extractor is trained with the same 18k tied-states, but using only the 309-hour Switchboard-I data. Using the features derived from this DNN, another GMM-HMM set is trained again on the full training set. Recognition performance of different systems is compared in Table 4. Comparing with the results in Table 1, the WER of the DNN-HMM trained on the same 309-hour data is reduced by 1.0% absolute (or 6.1% relative) because more nodes are used in the output layer (9.3k *vs.* 18k). The performance of the DNN-GMM-HMM system is improved significantly due to the above fact and using much more training data. Overall, an absolute WER reduction of 1.5% (or 9.8% relative) is achieved without unsupervised adaptation, and 1.6% absolute (or 10.9% relative) is achieved with adaptation.

If one could afford to wait the completion of training a DNN using the 2,000-hour full data set [29], even better performance can be achieved as shown in Table 5. The DNN trained using cross entropy criterion is used as the feature extractor for our DNN-GMM-HMM systems. The result reported in [30] by further fine-tuning the DNN-HMM with a sequence based MMI criterion is also included in Table 5 for comparison. It is observed that using more training data reduces the WER of DNN-HMM system moderately by 0.8% absolute (or 5.2% relative). It also improves the DNN feature extractor, therefore the DNN-GMM-HMM systems are improved compared with the ones using the feature extractor trained with 309-hour data. With tied-state based training using cross entropy criterion, both DNN-HMM and DNN-GMM-HMM systems achieve very close results (14.6% *vs.* 14.5%). After sequence-based MMI training, the DNN-GMM-HMM system achieves a WER of 13.0%, which is slightly better than the WER of 13.3% achieved by its DNN-HMM counterpart. Finally, after unsupervised adaptation, the WER of the DNN-GMM-HMM system is reduced to 12.3%. To the best of our knowledge, this is the best result reported on this benchmark evaluation.

3.3. Discussion

From the results in Tables 4 and 5, using the 2,000-hour trained DNN-HMM, the WER after MMI training (13.3%) is only 0.5% absolute (or 3.6% relative) better than the 309-hour trained DNN feature extractor combined with the 2,000-hour trained GMM-HMMs (13.8%). However, such a gain comes at a considerable cost in DNN training. Using a GPU implementation, it takes approximately 100 hours to perform a full data sweep on the 2,000-hour set using cross entropy criterion, and 170 hours using MMI criterion [30]. The whole training process needs to run several sweeps. In contrast, the GMM-HMM training algorithms can be easily parallelized to leverage large-scale CPU clusters. For example, our training tool only takes about several minutes to 3 hours for ML, WE-RDLT and MMI training, respectively, to perform a full data sweep on a cluster of 1,024 CPU cores.

On top of cross entropy criterion, sequence training of DNN-HMMs can bring additional WER reduction (e.g., [4,30]). A relative WER reduction of around 10% can usually be achieved, which is similar to what can be achieved by performing sequence training on GMM-HMMs. We are just wondering how sequence training would affect the DNN feature extractor as used in our approach, which could be one of our future experiments.

4. Conclusion

In this paper, we have demonstrated that using DNN-derived features for GMM-HMM based acoustic modeling can achieve similar or even better recognition accuracy than its DNN-HMM counterpart. The proposed approach combines the best of both worlds in deep learning and well-established GMM-HMM framework to offer a short-term practical solution. Our ongoing work include the investigation of different sampling strategies in the proposed approach and a truly scalable DNN training method to leverage large-scale training data for improved LVCSR accuracy. We will report those results elsewhere.

5. Acknowledgement

We would like to thank our colleagues Frank Seide and Gang Li for sharing their DNN-HMM training recipe and several DNNs used in this study, and Jinyu Li for his valuable feedbacks and discussions on this work.

6. References

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

[2] F. Seide, G. Li, and D. Yu, "Conversational speech transcription using context-depedent deep neural networks," in *Proc. InterSpeech-2011*, pp. 437–440.

[3] G. Hinton, L. Deng, D. Yu, G. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. Sainath, and B. Kingsbury, "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, Nov.

[4] B. Kingsbury, T. N. Sainath, and H. Soltau, "Scalable minimum Bayes risk training of deep neural network acoustic models using distributed Hessian-free optimization," in *Proc. InterSpeech-2012*, 4 pages.

[5] N. Jaitly, P. Nguyen, A. Senior, and V. Vanhoucke, "Application of pretrained deep neural networks to large vocabulary speech recognition," in *Proc. InterSpeech-2012*, 4 pages.

[6] Z.-J. Yan, Q. Huo, J. Xu, and Y. Zhang, "Tied-state based discriminative training of context-expanded region-dependent feature transforms for LVCSR," in *Proc. ICASSP-2013*, 5 pages.

[7] H. Hermansky, D. P. W. Ellis, and S. Sharma, "Tandem connectionist feature extraction for conventional HMM systems," in *Proc. ICASSP-2000*, pp. 1635–1639.

[8] F. Grézl, M. Karafiát, S. Kontár, and J. Černocký, "Probabilistic and bottle-neck features for LVCSR of meetings," in *Proc. ICASSP-2007*, vol. 4, pp. 757–761.

[9] D. Yu and M. L. Seltzer, "Improved bottleneck features using pretrained deep neural networks," in *Proc. InterSpeech-2011*, pp. 237–240.

[10] T. Sainath, B. Kingsbury, and B. Ramabhadran, "Auto-encoder bottleneck features using deep belief networks," in *Proc. ICASSP-2012*, pp. 4153–4156.

[11] Z. Tüske, M. Sundermeyer, R. Schlüter, and H. Ney, "Context-dependent MLPs for LVCSR: Tandem, hybrid or both?" in *Proc. InterSpeech-2012*, 4 pages.

[12] X. Chen, A. Eversole, G. Li, D. Yu, and F. Seide, "Pipelined back-propagation for context-dependent deep neural networks," in *Proc. InterSpeech-2012*, 4 pages.

[13] J. Dean, G. S. Corrado, R. Monga, K. Chen, M. Devin, Q. V. Le, M. Z. Mao, M. A. Ranzato, A. Senior, P. Tucker, K. Yang, and A. Y. Ng, "Large scale distributed deep networks," in *Proc. NIPS-2012*, 9 pages.

[14] N. Kumar and A. G. Andreou, "Heteroscedastic discriminant analysis and reduced rank HMMs for improved speech recognition," *Speech Communication*, vol. 26, no. 4, pp. 283–297, 1998.

[15] P. C. Woodland and D. Povey, "Large scale discriminative training of hidden Markov models for speech recognition," *Computer Speech and Language*, vol. 16, no. 1, pp. 25–47, 2002.

[16] M. J. F. Gales, "Maximum likelihood linear transformations for HMM-based speech recognition," *Computer Speech and Language*, vol. 12, pp. 75–98, 1998.

[17] 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-2013*, 5 pages.

[18] M. Seltzer, D. Yu, and Y.-Q. Wang, "An investigation of deep neural networks for noise robust speech recognition," in *Proc. ICASSP-2013*, 5 pages.

[19] C.-H. Lee and Q. Huo, "On adaptive decision rules and decision parameter adaptation for automatic speech recognition," *Proceedings of the IEEE*, vol. 88, no. 8, pp. 1241–1269, 2000.

[20] T. Anastasakos, J. McDonough, R. Schwartz, and J. Makhoul, "A compact model for speaker-adaptive training," in *Proc. ICSLP-1996*, pp. 1137–1140.

[21] Q. Huo and B. Ma, "Irrelevant variability normalization in learning HMM state tying from data based on phonetic decision-tree," in *Proc. ICASSP-1999*, pp. 577–580.

[22] Y. Zhang, J. Xu, Z.-J. Yan, and Q. Huo, "A study of an irrelevant variability normalization based discriminative training approach for LVCSR," in *Proc. ICASSP-2011*, pp. 5308–5311.

[23] J. Xu, Y. Zhang, Z.-J. Yan, and Q. Huo, "An i-vector based approach to acoustic sniffing for irrelevant variability normalization based acoustic model training and speech recognition," in *Proc. InterSpeech-2011*, pp. 1701–1704.

[24] D. Povey, B. Kingsbury, L. Mangu, G. Saon, H. Soltau, and G. Zweig, "fMPE: Discriminatively trained features for speech recognition," in *Proc. ICASSP-2005*, pp. 961–964.

[25] B. Zhang, S. Matsoukas, and R. Schwartz, "Discriminatively trained region dependent feature transforms for speech recognition," in *Proc. ICASSP-2006*, pp. 313–316.

[26] J. J. Godfrey, E. C. Holliman, and J. McDaniel, "SWITCHBOARD: Telephone speech corpus for research and development," in *Proc. ICASSP-1992*, pp. 517–520.

[27] C. Cieri, D. Miller, and K. Walker, "The Fisher corpus: A resource for the next generations of speech-to-text," in *Proc. 4th International Conference on Language Resources and Evaluation*, 2004, pp. 69–71.

[28] "The NIST scoring toolkit SCTK," see the following site for details: http://itl.nist.gov/iad/mig/tests/rt/2002/software.htm.

[29] G. Li, H. Zhu, G. Cheng, K. Thambiratnam, B. Chitsaz, D. Yu, and F. Seide, "Context-dependent deep neural networks for audio indexing of real-life data," in *Proc. SLT-2012*, pp. 143–148.

[30] H. Su, G. Li, D. Yu, and F. Seide, "Error back propagation for sequence training of context-dependent deep networks for conversational speech transcription," in *Proc. ICASSP-2013*, 5 pages.