SPEAKER ADAPTIVE TRAINING IN DEEP NEURAL NETWORKS USING SPEAKER DEPENDENT BOTTLENECK FEATURES

Rama Doddipatla

Toshiba Research Europe Limited, Cambridge Research Laboratory, Cambridge, UK

ABSTRACT

The paper proposes an approach to perform speaker adaptive training (SAT) in deep neural networks using a two-stage DNN. The first-stage DNN extracts speaker dependent bottleneck(SDBN) features by updating the weights of the BN layer with speaker specific data. Using the SDBN features, a second-stage DNN is trained in the SAT framework. Choosing the BN layer as the speaker dependent layer instead of one of the hidden layers reduces the number of parameters to be tuned using speaker specific data. Experiments are presented on the Aurora4 task, where the input features are normalised with constrained maximum likelihood linear regression (CMLLR) and speaker information is appended in the form of D-vectors. Following an unsupervised adaptation of BN layer, the proposed approach provides a relative gain of 8.6% and 8.9% WER on top of DNNs trained with FBANK features appended with and without D-vectors respectively. A relative gain of 10.3% WER is observed when applied on top of DNNs trained with CMLLR transformed FBANK features, but the gain in performance saturated when combined with D-vectors. It is observed that supervised adaptation with as little as one minute of audio from a specific speaker improved the performance when compared with the baseline.

Index Terms— Speaker adaptive training, speaker normalisation, deep neural networks, speaker dependent bottleneck features, automatic speech recognition.

1. INTRODUCTION

Speaker adaptation for DNNs is an active area of research and is shown to improve the performance of automatic speech recognition (ASR). A wide range of approaches have been proposed in literature, that can be broadly classified into two main categories based on where the speaker variability is normalised in DNNs. The normalisation can be applied either by transforming the feature space before training the DNN or modifying the parameters of an already trained DNN using data from a specific speaker.

Transforming the feature space before training the DNN, using feature transformations like vocal tract length normalisation (VTLN) [1] and constrained maximum likelihood linear regression (CMLLR) [2], have shown to improve the DNN performance. Appending speaker information to the input features, in the form of I-vector [3], D-vectors [4], vectorised CMLLR transforms [5] or using speaker codes [6, 7], to make the DNN aware of speaker specific changes have shown to improve the DNN performance. All these feature space transformations are applied before training the DNN. For approaches that tune the parameters of the DNN, the primary challenge is to tune the network parameters with limited amount of training data from a specific speaker. One of the ideas is to estimate speaker specific transformation by tuning weights of a specific layer, while the rest of the layers are kept fixed. These layers can be positioned either towards the front [8, 9], middle [9] or towards the output layer [10]. Other approaches looked at minimising the parameters by performing singular value decomposition (SVD) over DNN weights and estimate speaker dependent (SD) transformation inserted between the decomposed weight matrices [11]. In learning hidden unit contributions (LHUC), an SD vector is attached to every hidden layer learnt from the test speaker, which are applied to DNN hidden units with element wise multiplication [12]. Regularisation of the training with Kullback-Leibler (KL) divergence have also been studied to reduce over-fitting to the data [13] .

Compared with the approaches proposed for speaker adaptation in DNNs, there have been very few attempts to perform speaker adaptive training (SAT) for hybrid DNN-HMM systems, which is a well established approach in GMM-HMM systems. SAT performs speaker adaptation both in training and recognition [14, 15]. Training DNNs using features transformed with VTLN, CMLLR or appending speaker information in the form of speaker codes can be thought as training DNNs in the SAT framework. In [16], SAT training in DNNs is performed by allocating speaker dependent (SD) layers and tuning the weights of these layers using data from a specific speaker. Once the SD layers are tuned, the DNN is retrained with the SD layers to obtain the SAT-DNN model. In [17], SAT training is performed by linearly shifting the input features to a speaker normalised space before training the DNN. The linear shifts are estimated by transforming the I-vectors using a DNN.

In this paper, we propose an approach to perform speaker adaptive training (SAT) in DNNs for hybrid systems using a two-stage DNN architecture as proposed in [18]. The first-stage DNN is used for extracting bottleneck (BN) features, where speaker normalisation is applied by tuning the weights of the BN layer [19] to derive speaker dependent bottleneck (SDBN) feature. These speaker normalised features are used for training the second-stage DNN in the SAT framework. Experiments are conducted on the Aurora4 task, where the input features are normalised with CMLLR and appended with speaker information in the form of D-vectors. Following an un-supervised adaptation of the BN layer, it will be shown that the proposed approach provides a relative gain of 8.6% WER and 8.9% WER on top of DNNs trained with FBANK features appended with and without D-vectors respectively. The proposed approach when applied on top of DNNs trained with CMLLR transformed FBANK features provides a relative gain of 10.3% WER. The performance seems to saturate when CMLLR transformed FBANK features are combined with D-vectors. It will be shown that supervised adaptation of the BN layer with one minute of audio from a specific speaker provides improvement in performance when compared with the baseline.

The rest of the paper is organised as follows: first, the proposed approach to perform speaker adaptive training in DNNs is presented and our motivations for using a two-stage DNN framework, followed by the experimental setup, results and discussion. The performance

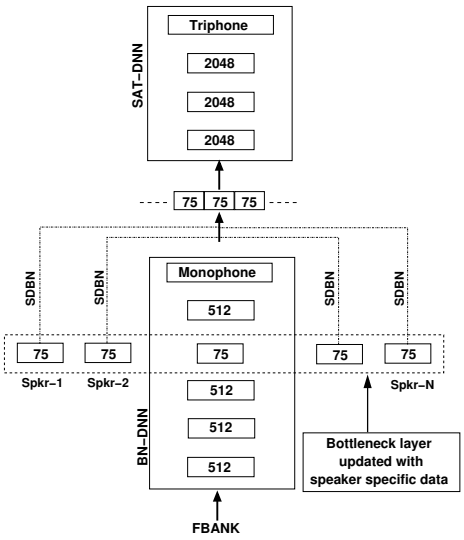

Fig. 1. Two stage DNN architecture used in the proposed scheme for speaker adaptive training.

of the proposed approach is studied both in supervised and unsupervised adaptation and finally present our conclusion and references.

2. SPEAKER ADAPTIVE TRAINING IN DNNS

Speaker adaptive training (SAT) is a well established approach to improve the acoustic model in GMM-HMM systems, where speaker adaptation is applied both during training and recognition. In order to perform speaker adaptive training in a hybrid DNN, a two-stage DNN architecture [18] as shown in Fig. 1 is used in the proposed approach. The first-stage DNN acts as a feature extractor, where speaker normalisation in performed on the bottleneck (BN) features by updating the weights of the bottleneck layer with data from a specific speaker. These features are in turn used for training the second-stage DNN to perform SAT.

Transforming the input features with CMLLR or appending speaker information in the form of I-vectors, D-vectors or speaker codes before training the DNNs already fall into the framework of speaker adaptive training. The proposed framework for SAT in DNNs not only allows us to perform similar speaker normalisation's on the front-end, but also facilitates us to rapidly adapt the parameters of the DNN using data from a specific speaker. Having such a framework help us integrate front-end normalisation with speaker adaptation approaches proposed to directly adapt the DNN model parameters.

The first-stage DNN is a bottleneck feature extractor (BN-DNN) trained using monophone targets. Speaker dependent bottleneck (SDBN) features are generated after tuning only the weights of the bottleneck layer with data from a specific speaker and keeping the weights in the rest to the layers fixed as proposed in [19]. Since the BN layer is considerably smaller in size when compared with the rest of the hidden layers, the number of parameters required for tuning with limited amount of training data from a specific speaker is also reduced. So, each speaker in the training or the recognition set will have a SDBN layer estimated using data from that speaker.

During training, once the SDBN layers are estimated for each speaker, the SDBN features are used for training the second-stage DNN to obtain the SAT-DNN model. During recognition, the SBDN layer is updated in an unsupervised approach using the previous pass ASR transcription. The reason for using monophone targets for training the BN-DNN is to make the SDBN layer training robust to transcription errors during recognition and alleviate the problem of data sparsity. A recent study looks at tying the output states by adding a soft-max layer of context independent states on top of the context dependent states to reduce the problem of data sparsity for updating the network parameters [20]. Mapping the triphone targets onto monophones can be interpreted as state tying and helps alleviate the problem of data sparsity. Since the BN-DNN is primarily employed only to extract BN features, it does not influence the performance of SAT-DNN. The number of targets in the hidden layers can also be reduced in order to reduce the number of parameter required for tuning using data from a specific speaker.

The summary of steps in both training and recognition for the proposed SAT in DNNs are as follows:

Training

• Train the BN-DNN using FBANK features and monophone targets. Optionally, the input features can be transformed with CMLLR and appended with speaker information in the form of speaker codes.

• Using the monophone alignments on the training data, tune the weights of the BN layer to extract SDBN features for each speaker.

• Using the SDBN features, train the second-stage DNN to obtain in the SAT-DNN. The BN features are spliced with 5 frames on either side for training the SAT-DNN.

Recognition

• Perform a first pass recognition using the network trained using FBANK features.

• Obtain the monophone alignments using the first-pass transcription and estimate the SDBN features for each of the test speaker by updating the weights of the BN layer.

• Using these SDBN features, perform recognition using the trained SAT-DNN model.

2.1. Relation to previous work

The proposed framework closely resembles the architecture proposed in [16], where speaker dependent (SD) layer is chosen to be one of the hidden layer in the DNN. Once the SD layers are trained using data from a specific speaker, the network is retrained to obtain the SAT-DNN. The SAT model seem to perform better when the SD layers were positioned in the middle of the network and regularisation is performed to avoid over-fitting. The SD layers in recognition are updated in an supervised approach, where the test data is divided into four sub-groups and recognition results are obtained in the four-times cross-validation scheme. In the proposed framework, the SD layer is chosen to be the BN layer instead of a hidden layer and is updated using monophone targets. Since the BN layer is much smaller in dimension when compared with the hidden layer, the number of parameter required for updating are also less. The SD layer is updated independently of the SAT-DNN rather than being part of the same network. More over the SD layers in recognition are updated in an unsupervised approach using previous pass ASR transcripts.

The proposed framework also has resemblance with the architecture proposed in [17], where I-vectors have been transformed to linear shifts using a DNN to perform speaker normalisation on the input features before training the SAT-DNN. The proposed framework also employs a first-stage DNN to process the input features to obtain speaker dependent bottleneck features that are used for training a second-stage DNN. In both the frameworks a DNN is employed to

Table 1. Comparing the performance of conventional 7 layer DNN with two-stage DNN architecture.

%WER	Conventional	Two-stage
FBANK	14.6	14.5
+ D-vec	13.9	13.9
+ CMLLR	12.6	12.6
+CMLLR + D-vec	12.3	11.9

perform speaker normalisation before training the SAT DNN.

3. EXPERIMENTS AND RESULTS

The ASR experiments in this paper are reported on the Aurora4 task. We will present a brief description of the corpus and then present the results on the proposed approach to perform SAT in DNNs.

3.1. Corpus description

The database is derived from WSJ corpus, in which additive noise and convolution distortion have been artificially added. The database is provided with clean and multi-condition training data having 7138 utterances from 83 speakers. The clean data is recorded with a primary Sennheiser microphone, where as the multi-condition database has data recorded with a primary as well as a secondary microphone which includes convolutive distortions. In all the experiments, the models are trained with multi-condition data, which includes the clean data as well as data having additive noise from six noise conditions, i.e. airport, babble, car, restaurant, street and train station. The test data consists of 330 utterances from 8 speakers, recorded by two different microphones, thus leading to 14 different test sets. We assume a well trained GMM-HMM system is already existing and will be presenting results of the DNN systems only . First we will compare the performance of a conventionally trained DNN that only has hidden layers with the two-stage DNN architecture used to perform the proposed SAT approach in the next section.

3.2. Conventional and two-stage DNN

The baseline DNN is trained using the conventional DNN architecture with 7 hidden layers having 2048 targets each. The input to the DNN uses 40 dimensional Mel filter-bank (FBANK) features spliced with 5 frames on either side to form a 440 dimensional feature vector as input. In the two-stage DNN, the BN-DNN uses 3 hidden layer with 512 targets each and trained using monophones as targets on the output layer. A bottleneck feature of 75 dimensions is extracted for each frame and spliced with 5 frames on either side to form a 825 dimension feature vector, which are used for training the second-stage DNN having 3 hidden layers with 2048 targets each. The output layer in both conventional and two-stage DNNs has 2281 targets, that are derived using alignments from the SAT GMM-HMM model. The DNNs in all cases are initialised with RBM pre-training and optimised using cross-entropy criterion. All the experiments are performed using the KALDI toolkit [21].

Speaker normalisation is performed on the input features by transforming the FBANK features with CMLLR transforms (CMLLR-FBANK) and appending speaker information in the form of D-vectors [4]. CMLLR transforms are estimated while training the SAT GMM-HMM model. D-vectors are obtained by training a bottleneck DNN with speaker labels as targets in the output layer. In our experiments, the D-vector is obtained by averaging the bottleneck features over an utterance and then appending the constant vector to the filter-bank features in an utterance. This means that the speaker representation for the same speaker is allowed to change across utterances from the same speaker.

Table 2. Results comparing the stage at which D-vectors are appended for performing speaker normalisation in the two-stage DNN architecture.

%WER	FBANK	BN
+ D-vec	13.9	13.8
+ CMLLR + D-vec	11.9	12.0

Baseline results comparing the conventional and two-stage DNN architectures are presented in Table 1. Unless specified, the results in the tables always report the average %WER over all the 14 test sets. The table shows how the performance progresses by performing speaker normalisation on the input features either by transforming with CMLLR or appending D-vectors (D-vec) or performing both the operations together. Both CMLLR and D-vectors seem to improve the ASR performance and the best performance is achieved when both the operations are performed together. The two-stage DNN seem to perform comparably and the results are inline with the performance observed in conventional DNN. It is interesting to observe that transforming the features with CMLLR or appending D-vectors, the BN features in the two-stage DNN seem to be speaker normalised and improve the ASR performance. This results indicate that BN-DNN can be used for integrating information from multiple sources and can facilitate to combine feature transformation on the front-end with approaches to tune the network parameters to normalise speaker variability. In the rest of the paper, the results of two-stage DNN are used as baseline for comparing the performance.

3.3. Appending the D-vector

D-vectors are extracted independent of the input features used for training the two-stage DNN and can be appended either at the input along with filter-bank features for training the BN-DNN or appended to the BN features before training the second-stage DNN. The motivation for this experiment is to understand which combination seem to be effective for performing speaker adaptation in the two-stage architecture. The results comparing the performance in both the configurations using D-vectors are presented in Table 2. One can observe that D-vectors seem to provide similar gains in performance either when they are appended to the FBANK features before extracting the BN features or when appended with the BN features for training the second-stage DNN. In all cases CMLLR is applied on the FBANK features. There is no gain in performance when they are appended both to the filter-bank features as well as the BN features. In all our experiments, D-vectors are always appended with the filter-bank features and the second-stage DNN is trained only using the BN features.

3.4. Speaker adaptive training

This section presents the results of the proposed approach to training the SAT-DNN. Speaker dependent bottleneck (SDBN) features are extracted after tuning the weights of the BN layer with data from a target speaker. During training the SDBN features are used for training the SAT-DNN model. During recognition, the weights of the BN layer are updated in an unsupervised approach using first-pass transcriptions and the SDBN features are used for recognition using the SAT-DNN model. Table 3 presents the result using the proposed training approach. The table presents how the performance of

Table 3. Results of the proposed SAT-DNN using SDBN features extracted from BN-DNN having 512 targets in the hidden layers.

%WER	Baseline	+ SAT-DNN	%WERR
FBANK	14.5	13.2	8.9
+ D-vec	13.9	12.7	8.6
+ CMLLR	12.6	11.3	10.3
+ CMLLR + D-vec	11.9	11.2	5.9

Table 4. Results of the proposed SAT-DNN using SDBN features extracted from BN-DNN having 256 targets in the hidden layers.

%WER	Baseline	+ SAT-DNN	%WERR
FBANK	15.6	14.4	7.7
+ D-vec	14.4	13.7	4.9
+ CMLLR	13.1	11.9	9.2
+ CMLLR + D-vec	12.0	11.4	5.0

SAT-DNN changes by applying front-end normalisation using CMLLR and D-vectors. We make the following observations:

• Using FBANK features, without any normalisation of the front-end, the proposed approach provides a relative gain of 8.9% in terms if word-error rate (WER) when compared with the baseline system.

• One can observe that as the baseline performance improves, the performance of SAT-DNN also gradually increases. Since an unsupervised approach is followed to update of the BN layer, improved transcriptions will also improve the SDBN features as there are less transcription errors.

• The biggest gain in performance is achieved when SAT is performed on top of DNN trained with CMLLR features, where we have a relative gain of 10.3% in WER.

• The performance seem to saturate and we notice very little gains in performance when SAT training is performed on top of DNNs trained with CMLLR features when combined with D-vectors.

3.5. Reducing the size of hidden layer in BN-DNN

Here we study the influence on reducing the number of targets in the hidden layers used for extracting the bottleneck features. The main motivation is to see if we can reduce the number of parameters for unsupervised adaptation of the BN layer without influencing the recognition performance. We perform experiments using hidden layer having 256 targets. The results are presented in Table 4. Comparing with the results using BN-DNN using 512 targets (in Table 3), one can observe that the baseline performance degrades and similarly the performance of SAT-DNN. An similar picture in terms of relative WER gains can be observed when SAT is performed on top of CMLLR-FBANK features. It is interesting to observe that using a smaller hidden layer with 256 targets in BN-DNN still could achieve a comparable performance when compared with a BN-DNN using a hidden layer with 512 targets. We have the impression that using larger size of hidden layers might help in extracting better BN features.

3.6. Supervised adaptation

The final set of experiments look at performing supervised adaptation of the BN layer using true transcripts available for the test speaker. The experiments are done on the BN-DNN having 512 targets in the hidden layers. Each speaker has 40 utterances and corresponds to having 5 minutes of audio data approximately. The BN layer weights are updated using 10, 20, 30 and all of the utterances available from the test speaker to see how the performance changes. Please note that the CMLLR transforms in these experiments are still estimated in the unsupervised approach and have not been re-estimated. The results are presented in Table 5.

Table 5. Results comparing the performance of supervised adaptation using variable number of utterances.

%WER	+10	+20	+30	+40
FBANK	13.4	12.7	12.3	11.9
+ D-vec	13.1	12.1	11.9	11.6
+ CMLLR	11.5	11.1	10.8	10.4
+ CMLLR + D-vec	11.4	10.8	10.5	10.4

One can notice that the performance improves as the amount of data from a specific speaker increases as expected. It is interesting to note that less amount of adaptation data is required to achieve similar or better performance when the data is already normalised using CMLLR and D-vectors than when compared with only using FBANK features. This might be because of a better acoustic model trained in the SAT framework. Comparing with the results presented in Table 3, as little as 10 utterances from each speaker can already improve the performance over the baseline, which correspond to approximately one minute of data from each speaker. One can also notice that performing SAT on top of CMLLR-FBANK features when combined with D-vectors seem to saturate and have a performance similar to only using CMLLR features. This might explain the behaviour observed in the unsupervised adaptation experiments presented in Table 3.

4. CONCLUSION

The paper proposed an approach to perform speaker adaptive training in DNNs using a two-stage DNN architecture. The first-stage DNN performed BN feature extraction to derive speaker dependent bottleneck features by updating the weights of the BN layer with speaker specific data. Using the speaker normalised bottleneck features, the second-stage DNN is trained in the SAT framework. We investigated the proposed approach to perform SAT in combination with transforming the input features with CMLLR and appending speaker information in the form of D-vectors. We showed that the two-stage DNN architecture has a performance similar to conventional DNN having 7 hidden layers. We also showed that the two-stage DNN had similar performance when the D-vectors are either appended to the FBANK features or appended to the BN features before training the second stage DNN.

The proposed SAT training provided a relative gain of 8.9% WER when applied on top of DNN trained using FBANK features and a relative gain of 10.3% WER when applied on top of DNN trained using CMLLR-FBANK features respectively. Though the proposed SAT training provided a relative gain of 8.6% WER on top of DNN trained with FBANK features and combined with D-vectors, the performance seem to saturate when SAT training is applied on top of DNN trained using CMLLR-FBANK features and combined with D-vectors. A behaviour also noticed when SAT training was applied in supervised adaptation on the test speakers. Supervised adaptation experiments showed that, updating the weights of the BN layer with as little as 10 utterances (correspond to approx. one minute of audio) can already improve the performance over the baseline. This suggests that the proposed approach can be used to rapidly adapt the DNN parameters to the test speaker with very little adaptation data.

5. REFERENCES

[1] R. Serizel and D. Giuliani, "Vocal tract length normalisation approaches to dnn-based children's and adults' speech recognition," in *Spoken Language Technology Workshop (SLT), 2014 IEEE*, Dec 2014, pp. 135–140.

[2] S. P. Rath, D. Povey, K. Veselý, and J. Cernocký, "Improved feature processing for deep neural networks.," in *Proc. of INTERSPEECH*, 2013.

[3] G. Saon, H. Soltau, D. Nahamoo, and M. Picheny, "Speaker adaptation of neural network acoustic models using i-vectors," in *Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on*, Dec 2013, pp. 55–59.

[4] Ehsan Variani, Xin Lei, Erik McDermott, Ignacio Lopez Moreno, and Jorge Gonzalez-Dominguez, "Deep neural networks for small footprint text-dependent speaker verification," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, 2014.

[5] Yun Tang, A. Mohan, R.C. Rose, and Chengyuan Ma, "Deep neural network trained with speaker representation for speaker normalization," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, May 2014, pp. 6329–6333.

[6] Shaofei Xue, O. Abdel-Hamid, Hui Jiang, Lirong Dai, and Qingfeng Liu, "Fast adaptation of deep neural network based on discriminant codes for speech recognition," *Audio, Speech, and Language Processing, IEEE/ACM Transactions on*, vol. 22, no. 12, pp. 1713–1725, Dec 2014.

[7] Yulan Liu, P. Karanasou, and T. Hain, "An investigation into speaker informed dnn front-end for lvcsr," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*, April 2015, pp. 4300–4304.

[8] Frank Seide, Gang Li, Xie Chen, and Dong Yu, "Feature engineering in context-dependent deep neural networks for conversational speech transcription," in *ASRU 2011*. December 2011, IEEE.

[9] B. Li and K. C. Sim, "Comparision of discriminative input and output transformation for speaker adaptation in the hybrid nn/hmm systems," in *Proc. of INTERSPEECH*, 2011.

[10] Kaisheng Yao, Dong Yu, Frank Seide, Hang Su, Li Deng, and Yifan Gong, "Adaptation of context-dependent deep neural networks for automatic speech recognition," in *in Proc. SLT12*, 2012.

[11] Jian Xue, Jinyu Li, Dong Yu, M. Seltzer, and Yifan Gong, "Singular value decomposition based low-footprint speaker adaptation and personalization for deep neural network," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, May 2014, pp. 6359–6363.

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

[13] Dong Yu, Kaisheng Yao, Hang Su, Gang Li, and Frank Seide, "Kl-divergence regularized deep neural network adaptation for improved large vocabulary speech recognition," in *ICASSP 2013*, 2013.

[14] Tasos Anastasakos, John Mcdonough, Richard Schwartz, and John Makhoul, "A compact model for speaker-adaptive training," in *in Proc. ICSLP*, 1996, pp. 1137–1140.

[15] T. Anastasakos, J. McDonough, and J. Makhoul, "Speaker adaptive training: a maximum likelihood approach to speaker normalization," in *Acoustics, Speech, and Signal Processing, 1997. ICASSP-97., 1997 IEEE International Conference on*, Apr 1997, vol. 2, pp. 1043–1046 vol.2.

[16] T. Ochiai, S. Matsuda, Xugang Lu, C. Hori, and S. Katagiri, "Speaker adaptive training using deep neural networks," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, May 2014, pp. 6349–6353.

[17] Yajie Miao, Hao Zhang, and Florian Metze, "Speaker adaptive training of deep neural network acoustic models using i-vectors," *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, vol. 23, no. 11, pp. 1938–1949, Nov. 2015.

[18] Jonas Gehring, Wonkyum Lee, Kevin Kilgour, Ian R Lane, Yajie Miao, and Alex Waibel, "Modular combination of deep neural networks for acoustic modeling," in *Proc. of INTERSPEECH*, 2013.

[19] Rama Doddipatla, Madina Hasan, and Thomas Hain, "Speaker dependent bottleneck layer training for speaker adaptation in automatic speech recognition," in *Proc. of INTERSPEECH*, 2014.

[20] R. Price, K.-I. Iso, and K. Shinoda, "Speaker adaptation of deep neural networks using a hierarchy of output layers," in *Spoken Language Technology Workshop (SLT), 2014 IEEE*, Dec 2014, pp. 153–158.

[21] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, Jan Silovsky, Georg Stemmer, and Karel Vesely, "The kaldi speech recognition toolkit," in *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*. Dec. 2011, IEEE Signal Processing Society.