ON COMBINING I-VECTORS AND DISCRIMINATIVE ADAPTATION METHODS FOR UNSUPERVISED SPEAKER NORMALIZATION IN DNN ACOUSTIC MODELS

Lahiru Samarakoon, Khe Chai Sim

School of Computing, National University of Singapore, Singapore

lahiruts@comp.nus.edu.sg, simkc@comp.nus.edu.sg

ABSTRACT

In automatic speech recognition (ASR), adaptation and adaptive training techniques are used to perform speaker normalization. Previous methods mainly focus on using these techniques in isolation. In contrast, this paper investigates two approaches to improve the ASR performance by combining i-vector based speaker adaptive training in deep neural network (DNN) acoustic models with discriminative adaptation techniques. First, we combine these techniques by interpolating the decoding lattices of i-vector based systems with the decoding lattices of a discriminatively adapted model. Then, we combine these methods by discriminatively adapting the i-vector based system in unsupervised fashion. Our experiments on TED-LIUM dataset show that compared with a strong speaker independent baseline, lattice interpolation and adaptation of the i-vector systems achieve 12.0% and 15.6% relative improvements, respectively. Moreover, in comparison to the i-vector based systems, lattice interpolation reported a 4.5% relative improvement while discriminatively adapting the i-vector system reported a 8.3% relative improvement.

Index Terms— Automatic speech recognition, deep neural networks, speaker normalization.

1. INTRODUCTION

In comparison to the conventional Gaussian mixture model (GMM) based systems, Deep neural network (DNN) based acoustic modeling has achieved state-of-the-art performance in ASR systems [1]. However, DNNs, like all other machine learning techniques, are susceptible to performance degradation due to the mismatch between the training and testing conditions. Normalization techniques transform the model to match the testing condition or augment the inputs to match the model. In ASR, speaker normalization techniques are used to minimize the mismatch between the training and testing conditions due to the speaker variability.

Maximum a posteriori (MAP) [2] and maximum likelihood linear regression (MLLR) [3] are commonly used to normalize GMM- hidden markov model (HMM) systems. In MAP, model parameters are re-estimated by maximizing the posterior probability. MLLR performs speaker normalization by estimating a linear transformation of the model parameters to reduce the speaker mismatch. An approach of combining these normalization techniques with superior feature representation learning power of DNNs is to train a tandem system [4, 5]. In tandem systems, a DNN is used to extract bottleneck features to train a GMM-HMM system.

The speaker normalization for DNNs is important as it improves the performance significantly [6–10]. However, due to the generative nature of GMMs, most of the conventional techniques cannot be directly used for discriminative DNNs. In addition, DNN-HMM systems have millions of parameters, which make most of the techniques prone to over-fitting, especially when the normalization should be performed with a small amount of data in unsupervised fashion.

In adaptation, test data is used to perform speaker normalization while in adaptive training, the training data is used to minimize the speaker variablity. The motivation behind our research is focused on performing speaker normalization using both training and test data by combining speaker adaptive training with adaptation techniques. In this paper, we propose to combine i-vector based speaker adaptively trained systems with discriminative adaptation techniques such as learning hidden unit contributions (LHUC) [11] and bias adaptation. We investigate this combination in two ways: first, by lattice interpolation and then by performing LHUC / bias adaptation on well-trained i-vector based systems. In addition, we compare the bias adaptation with LHUC method. Our experiments show that both techniques perform similarly when rectified linear units (ReLUs) are used as hidden units. LHUC requires modifications to the structure of the model. Therefore, if the DNN is trained with ReLUs, bias adaption can be performed without modifying the model structure. Furthermore, we investigate how these combinations of speaker normalization techniques perform when only a small amount of adaptation data is available, which is more congruent with real-world applications.

The rest of the paper is organized as follows. In Section 2, a brief review of the DNN speaker normalization techniques is given. Section 3 describes our experimental setup. Results are reported in Section 4 and we conclude our work in Section 5.

2. SPEAKER NORMALIZATION FOR DNN

Speaker normalization techniques for DNNs can be categorized into two broad approaches: adaptation, and adaptive training. Speaker adaptation methods deal with the speaker variability by changing a well-trained model to match the test speaker conditions, whereas speaker adaptive training learns a way to deal with the speaker mismatch during training.

Linear transformation based adaptation methods augment the original DNN model with a linear layer. Usually, the linear layer is initialized with an identity matrix and zero biases and is updated with the back-propagation (BP) algorithm using the adaptation data while keeping the weights of the original DNN fixed. The linear layer can be inserted between the input layer and the first hidden layer, known as linear input network (LIN) [12], or to the softmax layer known as linear output network (LON) [13] or between the hidden layers, known as a linear hidden network (LHN) [14]. Since adaptation of all the model parameters is more powerful, some methods adapt all the parameters by employing regularization into the adaptation criterion. In [7], a KL divergence based method is used to force the posterior distribution of the adapted model to be closer to that of the speaker independent (SI) model. In addition, the L_2 regularization [15] aims to keep the parameters of the adapted model closer to that of the SI model. However, to reduce the per-speaker footprint, some approaches perform the adaptation on a subset of parameters, including the last hidden layer [16], output layer biases [17], or more active hidden units of the network [16]. Another effective model adaptation technique is known as LHUC [11, 18], which learns speaker dependent hidden unit contributions during adaptation.

In DNN adaptive training, it is popular to provide speaker information with the acoustic features. The intuition behind this method is that a DNN is capable of exploiting the supplementary information about speakers to adjust the model parameters for speaker normalization. The i-vectors [6, 8, 9, 19] and bottleneck features [20] are commonly used as speaker representations. In addition, recently, cluster adaptive training (CAT) has been applied for speaker normalization [21, 22]. In CAT DNN approaches, a set of bases are estimated during training and followed by an interpolation vector estimation to combine the bases during testing. Another two ways of performing adaptive training on DNNs include, learning an adaptation network [23] and by spliting the DNN into speaker dependent and speaker independent layers [24]. Furthermore, the usage of CMLLR features for DNN training is also considered as adaptive training.

Since, the adaptation and adaptive training perform speaker normalization in two different ways, our goal is to combine state-of-the-art adaptation and adaptive training techniques to improve the ASR performance. Specifically, we focus on improving i-vector based speaker adaptive training by combining with LHUC and bias adaptation techniques.

3. EXPERIMENTAL SETUP

In this paper, all the experiments are performed on the first version of the TED-LIUM corpus [25]. The training set contains 118 hours of speech over 774 TED talks. In all our experiments, each talk is considered as a different speaker. We used 90% of the training set for training and the rest is used as the validation set. Our results are reported on the test set (tst2010) and the development set (dev2010) containing 11 and 8 speakers respectively.

First, MFCC features are extracted from speech using a 25-ms window and a 10-ms frame-shift. Cepstral mean normalization (CMN) per-speaker is then applied to the MFCCs. Linear discriminant analysis (LDA) features are obtained by first splicing 7 frames of 13-dimensional MFCCs and then projecting downwards to 40 dimensions using LDA. A global semi-tied covariance (STC) transformation [26] is applied on top of the LDA features. In addition, we apply a speaker specific constrained maximum likelihood linear regression (CMLLR) transform on top of the LDA features to create speaker normalized CMLLR features. The GMM-HMM system for generating the alignments for DNN baselines is built on top of these 40 dimensional CMLLR features.

All our DNNs have 6 sigmoid hidden layers with 2048 units per layer, and 4014 senones as the outputs. We trained our baselines on top of two different feature types, namely LDA and CMLLR. For each feature type, we trained two DNNs with ReLUs and sigmoid units for comparison. These baselines are trained on the acoustic features that span a context of 11 neighboring frames. Before being presented to the DNN, cepstral mean and variance normalization (CMVN) is performed on the features globally. To train the network, we use dropout pre-training with a rate of 0.5. All the DNNs are trained to optimize the cross-entropy criterion with a mini-batch size of 256 and a momentum of 0.9. For ReLUs we started DNN training with learning rate of 0.1 and adaptation performed with a large learning rate of 1.0 . For sigmoid DNNs, 1.0 and 5.0 learning rates were used for training and adaptation, respectively. All these learning rates are calculated per mini-batch. We used 3 iterations for all adaptation tasks. CNTK [27] is used to train the DNNs. The powerful Cantab language model [28] is used in decodings. The Kaldi toolkit [29] is used to build the GMM-HMM systems and for the i-vector extraction. The i-vectors are trained on top of the same 40 dimensional acoustic features (LDA or CMLLR). The universal background model (UBM) consist of 128 gaussians. We extracted i-vectors that are of 100 dimensions. In all our experiments, speaker-level i-vectors are used.

4. RESULTS

Table 1 shows results for various speaker normalization techniques on top of the LDA features. For the models with ReLUs, performances of bias adaptation and LHUC are very

Table 1. Word Error Rate (WER%) of various speaker normalization techniques on top of the SI LDA models.

Model	ReLU		Sigmoid	
	Test	Dev	Test	Dev
LDA	16.9	18.2	16.7	18.1
+ LHUC	15.5	17.3	15.3	17.1
+ Bias	15.6	17.3	15.5	17.4
+ i-vector	15.7	16.9	15.1	16.8

Table 2. WER (%) for various combinations of decoding lattice interpolations with a scale factor of 0.5.

Combination	ReLU		Sigmoid	
	Test	Dev	Test	Dev
i-vector & LHUC	15.0	16.8	14.7	16.5
i-vector & Bias	15.1	16.7	14.8	16.8
LHUC & Bias	15.5	17.3	15.4	17.2

similar. This is understandable since the bias shift and LHUC can operate in the same range for ReLUs. However, for sigmoid units, LHUC perform better than adapting the biases. This is because when sigmoids are used, the bias adaptation is restricted between the range [0,1]. Therefore, if the model is trained with ReLUs, it is possible to simply adapt the biases without adding extra parameters to learn amplitudes for LHUC. It is worth noting that LHUC is independent of the activation function. Aside from the improvements observed on the test set for the ReLUs model, the adaptive training using i-vectors reported the best WERs.

4.1. Techniques combination with lattice interpolation

For some speakers, i-vector system performed better and for the rest of the speakers LHUC / bias adaptation reported lower WERs. Therefore, we combined decodings by interpolating the lattices [30] with a scaling factor of 0.5. As it can be seen from Table 2, interpolating the lattices of i-vector system with the lattices of LHUC or bias adaptation improved the performance significantly. However, combining LHUC and bias adaptation lattices reported no improvements, which further indicates that LHUC and the bias adaptation are similar.

4.2. Techniques combination with discriminative adaptation

These findings led us to investigate the combination of i-vector based implicit speaker normalization with LHUC or bias adaptation by discriminatively adapting the speaker adaptively trained i-vector systems. From Table 3, it can

Table 3. WER (%) for discriminative adaptation of the adaptively trained i-vector systems.

Method	ReLU		Sigmoid	
	Test	Dev	Test	Dev
None	15.7	16.9	15.1	16.8
LHUC	14.7	16.4	14.2	16.3
Bias	14.7	16.4	14.4	16.6

Table 4. WER (%) for Bias adaptation analysis in i-vector systems.

Method	ReLU		Sigmoid	
	Test	Dev	Test	Dev
None	15.7	16.9	15.1	16.8
Bias	14.7	16.4	14.4	16.6
1^{st} layer Bias	14.9	16.4	14.6	16.2
update i-vec	14.9	16.4	14.7	16.3
+ LHUC	14.5	16.3	14.1	16.2

be seen that the combination of these techniques improved the performance significantly giving up to 15.0% and 9.9% relative improvements over the baseline systems for the test set and the developement set respectively. Moreover, up to 6.0% relative improvements were reported with reference to the best systems obtained when these techniques were used in isolation. The relative improvements on the development set is considerably lower than the that of the test set. This is a result of having more data per-speaker in the test set than the development set. The combination of LHUC and bias adaptation reported no improvements.

The i-vector based systems implicitly adapt the models by providing a bias shift to the first hidden layer. Therefore, it is worthwhile to investigate the adaptation of the first hidden layer biases. As can be seen in Table 4, adapting only the first hidden layer biases consistently improved the performance for both i-vector based DNN systems. This means that the i-vector based bias shift to the first hidden layer is not optimal and should be able to improve the performance by refining it. As given in Table 4, updating the i-vectors improved the performance. Furthermore, the gain achieved by adapting the i-vector is very similar to that of adapting only the first hidden layer bias. Adapting the i-vector has the advantage of having a low per-speaker footprint compared with the LHUC and bias adaptation methods. Since updating the i-vector only adapts the model in the first hidden layer, to adapt the model at all levels, we performed LHUC in combination with i-vector adaptation. As it can be seen in the last row of Table 4 , this reported the best performances on both datasets for DNNs trained with sigmoid units and ReLUs. In summary, with these combinations it was possible to obtain rel-

Table 5. WER (%) of various speaker normalization techniques on top of the CMLLR models.

Model	ReLU		Sigmoid	
	Test	Dev	Test	Dev
CMLLR	15.0	16.9	15.5	16.9
+ LHUC	14.4	16.4	14.7	16.7
+ Bias	14.5	16.3	14.8	16.8
+ i-vector	14.8	15.9	14.7	16.0
+ LHUC	14.2	15.7	14.0	15.7
+ Bias	14.2	15.8	14.2	15.8
+ update i-vec	14.3	15.8	14.2	16.0
+ LHUC	14.2	15.7	14.2	15.7

ative improvements upto 15.6% over the SI baselines and up to 6.6% relative improvements over the best performed technique when used in isolation.

Table 5 shows how these combinations performed on top of the CMLLR features. As it can be observed, absolute improvements up to 1.5% and 1.2% were reported on test and development sets, respectively. The improvements are smaller compared with that of the models trained on LDA features. This is simply because CMLLR features are already transformed to reduce the mismatch due to speaker variability.

4.3. Investigation with small amount of adaptation data

Next, we investigate how these techniques perform with a small amount of adaptation data. In this experiment, 48 seconds per-speaker is used on average. This was achieved by selecting only the first 5 segments from each test speaker for i-vector and CMLLR transform estimation. In addition, same amount of data is used for the adaptation alignment creation. We decided to select the data from the initial segments instead of selecting randomly to facilitate the reproducibility of these results. According to Table 6, both LHUC and bias adaptation decreases the WER by 2.4% relative. Both CMLLR and i-vector based adaptive training methods reported a relative improvement of 3.0%. Moreover, the combination of i-vectors based adaptive training with discriminative adaptation techniques reported the best results with relative improvements of 7.7%. However, the relative improvements are significantly lower compared with the self adaptation.

Finally, in Table 7, we performed the decoding lattice interpolation on the test set for the SI DNNs trained with ReLUs using 48 seconds of adaptation data per-speaker. As it can be seen, LHUC and bias adaptation combination recorded no improvement. As expected, combinations of i-vector based system with LHUC and bias adaptation improved the performance by 7.1% relatively. This is slightly lower than the improvements achieved from discriminatively adapting the i-vector based system as given in Table 6.

Table 6. WER (%) of various speaker normalization combinations when 48 seconds of adaptation data per-speaker is used in unsupervised fashion. Relative improvements over the baselines are given in brackets.

Method	LDA SI baseline	LDA i-vector system
None	16.9 (-)	16.4 (3.0)
+ LHUC	16.5 (2.4)	15.6 (7.7)
+ Bias	16.5 (2.4)	15.6 (7.7)
+ Update i-vec	-	15.6 (7.7)
+ CMLLR	16.4 (3.0)	-

Table 7. WER (%) for various combinations of decoding lattice interpolations when 48 seconds of adaptation data per-speaker is used in unsupervised fashion. Relative improvements over the baselines are given in brackets.

Combination	WER
i-vector & LHUC	15.7 (7.1)
i-vector & Bias	15.7 (7.1)
LHUC & Bias	16.5 (2.4)

5. CONCLUSIONS

In this paper, we investigated two ways of combining i-vector based adaptive training with unsupervised discriminative adaptation techniques for speaker normalization in DNNs. Firstly, we interpolated decoding lattices of an i-vector based system with the decoding lattices of a model, after the model was discriminatively adapted using LHUC or bias adaptation techniques. The lattice interpolation reported up to 12.0% and 4.5% relative improvements over the SI baseline and the i-vector based system, respectively. Secondly, we showed that by discriminatively adapting i-vector systems, it is possible to achieve upto 15.6% and 8.3% relative improvements over the SI baseline and the i-vector based system, respectively. In addition, we presented the results for these combinations on DNNs trained using speaker normalized CMLLR features. Furthermore, we empirically showed that when ReLUs are used, bias adaptation has the same effect as the LHUC adaptation. Finally, we investigated how these combinations perform when the amount of available adaptation data is small.

6. ACKNOWLEDGMENTS

This research was partially supported by the Google Faculty Research Award.

7. REFERENCES

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

[2] J. Gauvain and C.H. Lee, "Maximum a posteriori estimation for multivariate gaussian mixture observations of markov chains," *IEEE Transactions on Speech and Audio Processing*, vol. 2, no. 2, pp. 291–298, 1994.

[3] C. J. Leggetter and P.C. Woodland, "Maximum likelihood linear regression for speaker adaptation of continuous density hidden markov models," *Computer Speech & Language*, vol. 9, no. 2, pp. 171–185, 1995.

[4] H. Hermansky, D.P.W. Ellis, and S. Sharma, "Tandem connectionist feature extraction for conventional HMM systems," in *ICASSP*. IEEE, 2000, pp. 1635–1638.

[5] P. Bell, P. Swietojanski, and S. Renals, "Multi-level adaptive networks in tandem and hybrid ASR systems," in *ICASSP*. IEEE, 2013, pp. 6975–6979.

[6] L.T. Samarakoon and K.C. Sim, "Learning factorized transforms for speaker normalization," in *ASRU*. IEEE, 2015.

[7] D. Yu, K. Yao, H. Su, G. Li, and F. Seide, "Kl-divergence regularized deep neural network adaptation for improved large vocabulary speech recognition," in *ICASSP*. IEEE, 2013, pp. 7893–7897.

[8] V. Gupta, P. Kenny, P. Ouellet, and T. Stafylakis, "I-vector-based speaker adaptation of deep neural networks for french broadcast audio transcription," in *ICASSP*. IEEE, 2014, pp. 6334–6338.

[9] G. Saon, H. Soltau, D. Nahamoo, and M. Picheny, "Speaker adaptation of neural network acoustic models using i-vectors," in *ASRU*. IEEE, 2013, pp. 55–59.

[10] O. Abdel-Hamid and H. Jiang, "Fast speaker adaptation of hybrid NN/HMM model for speech recognition based on discriminative learning of speaker code," in *ICASSP*. IEEE, 2013, pp. 7942–7946.

[11] P. Swietojanski and S. Renals, "Learning hidden unit contributions for unsupervised speaker adaptation of neural network acoustic models," in *SLT*. IEEE, 2014, pp. 171–176.

[12] V. Abrash, H. Franco, A. Sankar, and M. Cohen, "Connectionist speaker normalization and adaptation," in *Eurospeech*. ISCA, 1995, pp. 2183–2186.

[13] B. Li and K. C. Sim, "Comparison of discriminative input and output transformation for speaker adaptation in the hybrid nn/hmm systems," in *INTERSPEECH*. ISCA, 2010, pp. 526–529.

[14] R. Gemello, F. Mana, S. Scanzio, P. Laface, and R. D. Mori, "Adaptation of hybrid ANN/HMM models using linear hidden transformations and conservative training," in *ICASSP*. IEEE, 2006, pp. 1189–1192.

[15] X. Li and J. Bilmes, "Regularized adaptation of discriminative classifiers," in *ICASSP*. IEEE, 2006, vol. 1, pp. I–I.

[16] J. Stadermann and G. Rigoll, "Two-stage speaker adaptation of hybrid tied-posterior acoustic models," in *ICASSP*. IEEE, 2005, pp. 977–980.

[17] K. Yao, D. Yu, F. Seide, H. Su, L. Deng, and Y. Gong, "Adaptation of context-dependent deep neural networks for automatic speech recognition," in *SLT*. IEEE, 2012, pp. 366–369.

[18] P. Swietojanski and S. Renals, "Differentiable pooling for unsupervised speaker adaptation," in *ICASSP*. IEEE, 2015, pp. 4305–4309.

[19] A. Senior and I. Lopez-Moreno, "Improving dnn speaker independence with i-vector inputs," in *ICASSP*. IEEE, 2014, pp. 225–229.

[20] H. Huang and K. C. Sim, "An investigation of augmenting speaker representations to improve speaker normalization for DNN-based speech recognition," in *ICASSP*. IEEE, 2015, pp. 4610–4613.

[21] T. Tian, Q. Yanmin, Y. Maofan, Z. Yimeng, and K. Yu, "Cluster adaptive training for deep neural network," in *ICASSP*. IEEE, 2015, pp. 4325–4329.

[22] C. Wu and M. J. F. Gales, "Multi-basis adaptive neural network for rapid adaptation in speech recognition," in *ICASSP*. IEEE, 2015, pp. 4315–4319.

[23] Yajie M., Hao Z., and Florian M., "Towards speaker adaptive training of deep neural network acoustic models," in *INTERSPEECH*. ISCA, 2014.

[24] O. Toshihiko, M. Shodai, L Xugang, H. Chiori, and K. Souichi, "Speaker adaptive training using deep neural networks," in *ICASSP*. IEEE, 2014, pp. 6349–6353.

[25] A. Rousseau, P. Deléglise, and Y. Esteve, "Ted-lium: an automatic speech recognition dedicated corpus.," in *LREC*. ELRA, 2012, pp. 125–129.

[26] M.J.F. Gales, "Semi-tied covariance matrices for hidden markov models," *IEEE Transactions on Speech and Audio Processing*, vol. 7, no. 3, pp. 272–281, 1999.

[27] D. Yu, A. Eversole, M. Seltzer, K. Yao, Z. Huang, B. Guenter, O. Kuchaiev, Y. Zhang, F. Seide, H. Wang, et al., "An introduction to computational networks and the computational network toolkit," Tech. Rep., Tech. Rep. MSR, Microsoft Research, 2014, http://codebox/cntk, 2014.

[28] W. Williams, N. Prasad, D. Mrva, T. Ash, and T. Robinson, "Scaling recurrent neural network language models," in *ICASSP*. IEEE, 2015, pp. 5391–5395.

[29] D. Povey, A. Ghoshal, G. Boulianne, N. Goel, M. Hannemann, Y. Qian, P. Schwarz, and G. Stemmer, "The kaldi speech recognition toolkit," in *ASRU*. IEEE, 2011.

[30] D. Povey, M. Hannemann, G. Boulianne, L. Burget, A. Ghoshal, M. Janda, M. Karafiát, S. Kombrink, P. Motlicek, Y. Qian, et al., "Generating exact lattices in the wfst framework," in *ICASSP*. IEEE, 2012, pp. 4213–4216.