ONLINE SPEAKER ADAPTATION FOR LVCSR BASED ON ATTENTION MECHANISM

Jia Pan*, Diyuan Liu†, Genshun Wan†, Jun Du*, Qingfeng Liu†, Zhongfu Ye*
* National Engineering Laboratory for Speech and Language Information Processing
University of Science and Technology of China, Hefei, Anhui, China
E-mail: jiapan@iflytek.com, {jundu, yezf}@ustc.edu.cn
†iFlytek Research, iFlytek Co., Ltd., Hefei, Anhui, P. R. China
E-mail: {dyliu2, gswan, qfliu}@iflytek.com

Abstract— **Speaker adaptation is one of the most popular and important topics for speech recognition. In this paper, we propose a novel online speaker adaptation technique for deep neural networks based large vocabulary automatic speech recognition (LVCSR). In this approach, the i-vectors of the speakers in training set are extracted as a static memory. For each frame, attention mechanism is used to select the most relevant speaker i-vectors to the current speech segment from the memory. We also propose a new attention mechanism to improve the performance. The vectors obtained by the attention mechanism provide speaker information for improving the accuracy of speech recognition. Experiments on the Switchboard task show that the proposed approach achieves a relative 8.3% word error rate (WER) reduction over speaker independent model without any adaptation data. The result is comparable to that of the popular i-vector based offline speaker adaption method and is much better than that of the i-vector based online speaker adaption method.**

I. Introduction

Recently deep learning based acoustic models [1] such as recurrent neural networks [2] and convolutional neural networks [3] have become the dominant acoustic modeling approach for automatic speech recognition (ASR) due to the superior accuracy over traditional Gaussian mixture model (GMM) based systems. In most cases they can achieve good performance by utilizing tremendous amount of training data, but they still suffer from obvious performance degradation when tested in mismatched conditions such as unseen speakers and environments. Over the past years, several speaker adaptation techniques have been proposed to solve this problem and achieved some success.

One way to do speaker adaptation is using auxiliary features. In [4], [5] and [6], i-vectors or bottleneck vectors are extracted from speaker recognition task and concatenated with acoustic features to provide speaker characteristics for the acoustic model. Abdel-Hamid et al. [7], [8], [9] proposed speaker code to represent speaker characteristics and jointly learned speaker code with acoustic model. Another way of speaker adaptation is model space adaption. In [10], [11] and [12], additional layers are appended to neural networks and the parameters of these layers are tuned by adaptation data while keeping the weights of the other layers fixed. Furthermore to avoid overfitting, conservative training methods are proposed. For example, Yu et al. [13] used Kullback-Leibler divergence (KLD) regularization to keep the weights of speaker dependent (SD) model not far from that of speaker independent (SI) model. In [14], it's found that only parts of the weights of recurrent neural networks should be retrained to reduce the number of parameters to be adapted. Multi-task learning strategy is adopted for speaker adaptation in [15] to suppress the influence of recognition errors.

These speaker adaptation methods can achieve considerable performance improvement over SI model when a number of supervised or unsupervised adaptation data is provided. However, in real-world LVCSR tasks such as short message dictation task running on Siri or iFlytek Voice Input, adaptation data is not easy to obtain. In [16], the authors utilize the click-through data of users to do speaker adaptation. But in most cases there is no enough adaptation data, especially for new speakers. In [17], several DNNs are trained for different speaker clusters in advance and a combination coefficient is learned using the online data. But the adaptation process need two-pass recognition and many multiples more storage space is necessary. To the best of our knowledge there are no fast and efficient online speaker adaptation method used in any practical systems yet.

In this paper, we proposed an attention based online speaker adaption method for LVCSR. The i-vectors of the speakers in training set are obtained as a static memory in advance. The similarity of a speech segment and each speaker i-vector is modeled by attention mechanism and learned jointly with the acoustic model from a large amount of training data, so that we can choose the closest speaker i-vectors to improve the recognition accuracy. Another advantage of the method is that adaptation can be done during one-pass recognition because that the recognition result is not needed in the method. Results on the Switchboard task show that the proposed approach achieves a relative 8.3% WER reduction over SI model. The results tell us that the proposed method is comparable to the i-vector based offline speaker adaption method in which extra information of other test utterances is used, and much better than the i-vector based online speaker adaption method.

The remainder of the paper is organized as follows. We introduce the i-vector technique and our proposed method in Section II and Section III. Experiments and results are described in Section IV. Section V concludes this paper and gives the future work.

II. I-VECTOR TECHNIQUE

In our study, the i-vectors of the speakers in training set are extracted as a static memory, so we introduce the i-vector technique firstly. The i-vector approach was first introduced for speaker recognition [18]. In this method, a large GMM with K diagonal covariance Gaussians which is known as the universal background model (UBM) is trained at first to represent the distribution of acoustic features, denoted as

$$x_t \sim \sum_{k=1}^{K} c_k N(\mu_k, \Sigma_k). \quad (1)$$

In (1), μ_k and Σ_k represent the mean and the covariance of the k-th Gaussian respectively, c_k is the weight of the k-th Gaussian and x_t represents the acoustic feature vector of frame t. Then the means of the UBM is adapted for each speaker as the following expression.

$$M_s = m + Tw_s \quad (2)$$

In (2), m is the speaker independent and channel independent super-vector which is obtained by splicing the mean vectors of all the Gaussian components of the UBM together. T is a matrix of low rank, called the loading matrix, and w_s is a random vector having a standard normal distribution. w_s is deemed as the identity vector (i-vector) for speaker s. T and w_s are learned jointly using a Maximum A Posteriori (MAP) criterion. Normally, LDA is carried out subsequently to reduce the dimension and improve the discrimination of i-vectors.

III. THE PROPOSED METHOD

A. Model Structure

Fig. 1 illustrates the structure of the proposed method in this paper. The structure consists of three parts, namely the memory block, the main network and the attention block. The memory block is made of the i-vectors of the speakers in training set. The i-vectors are extracted as the method introduced in Section II and clustered to K classes subsequently. Each item of the memory is an i-vector representing the characteristics of one cluster of speakers. The memory is denoted as $M = \{\boldsymbol{m}_1, \boldsymbol{m}_2, \dots, \boldsymbol{m}_K\}$, in which $\boldsymbol{m}_k$ represents the k-th item.

The main network can be any type of deep neural networks used for acoustic modeling, including feedforward neural networks (FFNNs), convolutional neural networks (CNNs) or recurrent neural networks (RNNs). CNNs and RNNs are preferred because of their excellent ability of sequence representation. Given a speech segment having T frames, the acoustic features are represented by $X = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_T\}$, where each $\boldsymbol{x}_t$ represents the feature vector at frame t. The corresponding outputs of the l-th hidden layer of the main network are denoted as $H^l = \{\boldsymbol{h}_1^l, \boldsymbol{h}_2^l, \dots, \boldsymbol{h}_T^l\}$.

For each hidden layer and each frame, the attention mechanism [19] is applied to select the most relevant items from the memory. In the attention block, a temporal pooling layer is built on top of each hidden layer to collect the segmental level information. The detail of the temporal pooling layer is described as follows.

$$\boldsymbol{c}_t^l = \sum_{i=1}^{t-1} \boldsymbol{h}_i^l \quad (3)$$

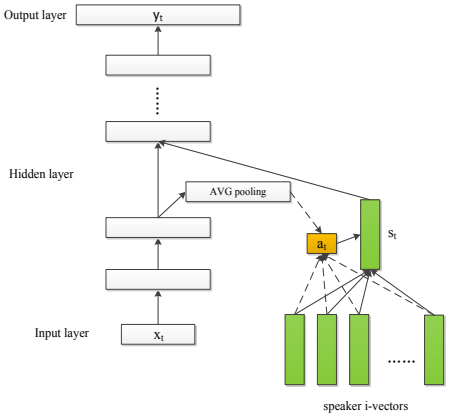

Fig. 1. The structure of the attention based online speaker adaptation. Note that the attention mechanism is described only for the second hidden layer for simplicity.

In (3), $\boldsymbol{c}_t^l$ is the output vector of the l-th temporal pooling layer at frame t, which is the average of the outputs of the l-th hidden layer ranging from frame 1 to frame t-1. With t going up, more long-time information such as speaker and channel information is represented by $\boldsymbol{c}_t^l$ and the phonemic info becomes blurred. An attention model taking $\boldsymbol{c}_t^l$ and $\boldsymbol{m}_k$ as the inputs is then built to learn the similarity scores between the two input vectors. The attention model is usually implemented through a Multi-Layer Perceptron (MLP) which is defined by the following expression.

$$e_{t,i}^l = \boldsymbol{v}^l \tanh(W^l \boldsymbol{c}_t^l + U^l \boldsymbol{m}_i) \quad (4)$$

In (4), $e_{t,i}^l$ is the attention value scoring the similarity between $\boldsymbol{c}_t^l$ and $\boldsymbol{m}_i$, the matrices W^l, U^l and the vector $\boldsymbol{v}^l$ are parameters of the model. The attention values are normalized through a softmax operation:

$$a_{t,i}^l = \exp(e_{t,i}^l) / \sum_{j=1}^{K} \exp(e_{t,j}^l). \quad (5)$$

The normalized attention values $\boldsymbol{a}_t^l$ are used to compute a summary of the items in the memory by:

$$\boldsymbol{s}_t^l = \sum_{i=1}^{K} a_{t,i}^l \boldsymbol{m}_i. \quad (6)$$

The speaker embedding vector $\boldsymbol{s}_t^l$ is then attached to $\boldsymbol{h}_t^l$, the resulting new vector $\overline{\boldsymbol{h}}_t^l = [\boldsymbol{h}_t^l \ \boldsymbol{s}_t^l]^T$ is used to calculate $\boldsymbol{h}_t^{l+1}$. It's noted that $\overline{\boldsymbol{h}}_t^l$ is only used for the calculation of $\boldsymbol{h}_t^{l+1}$ and should not be used for the calculation of $\boldsymbol{h}_{t+1}^l$.

B. Improvements of the attention mechanism

The attention values are normally normalized through the softmax operation. But when the number of the items in the memory increases, the softmax operation has the probability of information loss because the generated attention values are always sparse. To avoid the problem, similar to [20], we introduce sigmoid attention by replacing the softmax function in (5) with the logistic sigmoid function:

$$a_{t,i}^l = 1/(1 + \exp(-e_{t,i}^l)). \quad (7)$$

Because of the short-time stationarity of speech signals, the attention values should not change greatly between consecutive frames. To utilize this knowledge, RNN attention method is proposed and illustrated in Fig. 2. In this method an additional term is added to the calculation formula of the attention values:

$$e_{t,i}^{l} = \boldsymbol{v}^{l} \tanh\left(W^{l}\boldsymbol{c}_{t}^{l} + U^{l}\boldsymbol{m}_{i} + \sum_{k=1}^{\tau} \boldsymbol{g}_{k} a_{t-k,i}^{l}\right). \quad (8)$$

In (8) τ is the window size showing how many previous frames are involved, the vector $\boldsymbol{g}_k$ is the parameters of the model. The additional term will constrain the attention values not far away from the previous ones.

C. Training and inference

The whole model is trained jointly with the frame-level cross entropy criterion:

$$L_{CE} = \sum_{m=1}^{M} \sum_{t=1}^{T_m} p\left(y_{l_t^m} | \boldsymbol{x}_t^m\right). \quad (9)$$

In (9) $\boldsymbol{x}_t^m$ is the acoustic feature vector at frame t in utterance m and l_t^m is the corresponding state label.

The memory should not be updated during the training step. To match the test condition, the speaker label of each utterance in the training set is not used during training. During inference, none of the model parameters is updated because of the lack of adaptation data. The speaker embedding vector is obtained by the attention mechanism for each frame in the test utterance. To improve the efficiency, alternatively, we can get the speaker embedding vector every k frames, keeping the vector invariant in every k frames.

IV. Experiments

A. Experimental setup

The proposed approach is evaluated on the Switchboard (SWB) task. The training data consists of 20 hour Call Home English training set and 309-hour Switchboard-I training set containing total 5110 speakers. We use the SWB part of the NIST 2000 Hub5 evaluation set as the test set, containing 1831 utterances from 40 speakers. The 13-dimensional PLP features with their first and second order derivatives are extracted to train a GMM-HMM ASR model, which is used to obtain the state-level alignments for training deep neural networks. These features are pre-processed with speaker based mean and variance normalization. The cross-word tri-phone GMM-HMM model with 8991 tied-states and 360k Gaussians is trained with maximum likelihood criterion. A trigram language model is trained on the 2000h Fisher-corpus transcripts with the 39k dictionary for test.

B. Baseline systems

The SI baseline is a unidirectional LSTM model trained with the frame-level cross entropy criterion. The inputs of the model are the 40-dimensional log Mel-scale filter-bank outputs processed with mean and variance normalization. On top of the input layer there are 3 stacked LSTM layers with projection, each layer has 2048 memory cells and 512 output units. We delay the output HMM state label by 8 frames to use future frames to help LSTM make better predictions for current frame.

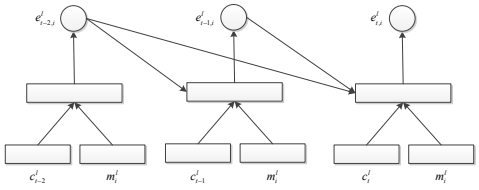

Fig. 2. The structure of the RNN attention mechanism. Note that τ is set to 2 for simplicity.

We stack 32 utterances in parallel in a mini-batch to speed up training. All experiments in our paper are conducted using the CAFFE toolkit and run on a server equipped with 4 Tesla P40 GPUs.

The SD baseline is also a unidirectional LSTM model trained with the frame-level cross entropy criterion. The structure of the model is the same as the SI model, except that the input features are augmented with speaker i-vectors. A 512 diagonal component UBM is trained first using all the training data and the 39-dimensional PLP features mentioned above. Then 300-dimensional speaker i-vectors are extracted and further compressed to 64 dimension by LDA followed by length normalization. The SI model is used to initialize the SD model to speed up the convergence. We test the i-vector based SD model in both online and offline manner. In the online manner, sentence-level i-vectors are extracted for each utterance in the test set separately, while in the offline manner speaker-level i-vectors are extracted using all the test utterances of each speaker, about 45 utterances for each speaker. During testing each utterance uses the sentence-level i-vector in the online manner and the speaker-level i-vector of the corresponding speaker in the offline manner, respectively.

Table I gives the results of the SI baseline model and SD baseline model. When tested in the offline manner, the WER of the SD model is 15.6%, a relative 7.7% WER reduction over the SI model, the WER of which is 16.9%. But when tested in the online manner, the WER is 16.3%. Comparing to the offline manner, only half of the WER reduction can be achieved because that the sentence-level i-vector cannot represent the speaker information very well.

C. Results of our method

We first evaluate the proposed online speaker adaptation approach. The main network of our speaker adaptation model is the same as the baseline model which consists of 3 stacked LSTM layers. The i-vectors of the speakers in the training set are clustered to 128 classes by the K-means algorithm to speed up training. The vector obtained by the softmax attention mechanism is concatenated only to the output of the last hidden layer because we find that the speed of training is considerably slowed down if we use the attention mechanism for all the hidden layers. The weights of the first two LSTM layers are initialized by the SI baseline model while the other parameters of the model are randomly initialized. We keep the learning rate and other training strategies consistent with the baseline model training for a fair comparison.

The results given in Table I show that a relative 6.5% WER

TABLE I
PERFORMANCE COMPARISON ON THE TEST SET. THE SOFTMAX BASED ATTENTION MECHANISM IS USED IN OUR PROPOSED METHOD.

Methods	WER (%)	Relative WER reduction over SI baseline
SI baseline	16.9	0%
i-vector based SD baseline tested in offline manner	15.6	7.7%
i-vector based SD baseline tested in online manner	16.3	3.6%
The proposed method	15.8	6.5%

TABLE II
RESULTS OF THE IMPROVEMENT METHODS.

Methods	WER (%)	Relative WER reduction over SI baseline
128 clusters	15.8	6.5%
256 clusters	15.8	6.5%
256 clusters + sigmoid attention	15.7	7.1%
256 clusters lusters + sigmoid attention + RNN attention	15.5	8.3%

reduction is achieved over the SI baseline model. The performance is much better than that of the popular i-vector based online speaker adaptation method and only slightly worse than the result of offline speaker adaptation which uses extra information of other test utterances. It implies that the speaker i-vectors closest to the current speech segment can be found by the attention mechanism.

To further improve the performance, we increase the number of the clusters of i-vectors from 128 to 256. The result given at line 3 in Table II shows that no gain is achieved. It implies that the softmax attention may cause information loss due to the sparse attention values. The guess is proved by the result given at line 4 in Table II. When we replace the softmax attention by the sigmoid attention, 0.1% absolute gain is achieved. On this basis, the RNN attention is executed to model the relationship between the attention values of consecutive frames. An extra 0.2% absolute WER reduction is achieved, that is shown at line 5 in Table II. In total, a relative 8.3% WER reduction over the SI baseline is achieved. The result is a little better than the i-vector based offline speaker adaptation method and much better than the i-vector based online speaker adaption method.

V. Conclusions

In this study, we have proposed an attention based online speaker adaptation method for deep neural networks based LVCSR. We find that the attention mechanism can align the speech segment to the corresponding speakers well. The results on the Switchboard task show that our proposed approach can improve WER over the SI model by up to 8.3% relative, which is comparable to the performance with the i-vector based offline speaker adaptation method and much better than the i-vector based online speaker adaptation method. Moving forward, we want to verify our method on larger tasks, and we plan to apply this method for offline speaker adaptation with a number of adaptation data.

Acknowledge

This work was supported in part by the National Key R&D Program of China under contract No. 2017YFB1002202, the National Natural Science Foundation of China under Grants No. 61671422 and U1613211, the Key Science and Technology Project of Anhui Province under Grant No. 17030901005, MOE-Microsoft Key Laboratory of USTC, and Huawei Noah's Ark Lab.

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. Audio, Speech, Lang. Process.*, vol. 20, no. 1, pp. 3042, 2012.

[2] H. Sak, A. Senior, and F. Beaufays, "Long short-term memory recurrent neural network architectures for large scale acoustic modeling," *Proc. of Interspeech*, 2014.

[3] T. Sercu, C. Puhrsch,B. Kingsbury, Y. Lecun, "Very deep multilingual convolutional neural networks for LVCSR," *Proc. of ICASSP*, 2016.

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

[5] P. Karanasou,Y. Wang, M. J. F. Gales, "PC Woodland Adaptation of Deep Neural Network Acoustic Models Using Factorised I-Vectors," *Proc. of Interspeech*, 2014.

[6] P. Cardinal, N. Dehak, Y. Zhang, J. Glass, "Speaker Adaptation Using the I-Vector Technique for Bottleneck Features," *Proc. of Interspeech*, pp. 2867-2871, 2015.

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

[8] S. Xue, O. Abdel-Hamid, H. Jiang, L. Dai, and Q. Liu, "Fast adaptation of deep neural network based on discriminant codes for speech recognition," *IEEE/ACM Trans. on Audio, Speech and Lang. Proc.*, vol. 22, no. 12, pp. 1713–1725, 2014.

[9] S. Xue, O. Abdel-Hamid, H. Jiang, L. Dai, "Direct adaptation of hybrid DNN/HMM model for fast speaker adaptation in LVCSR based on speaker code," *Proc. of ICASSP*, 2014.

[10] Z. Wang, D. Wang, "Unsupervised speaker adaptation of batch normalized acoustic models for robust ASR," *Proc. ICASSP*, pp. 4890-4894, 2017.

[11] P. Swietojanski, S. Renals, "Learning hidden unit contributions for unsupervised speaker adaptation of neural network acoustic models," *Spoken Language Technology Workshop*, pp. 171-176, 2015.

[12] L. Samarakoon, B. Mak, K. Sim, "Learning effective factorized hidden layer bases using student-teacher training for LSTM acoustic model adaptation," *Proc. of ICASSP*, pp. 5954-5958, 2018.

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

[14] C. Liu, Y. Wang, "Investigations on speaker adaptation of LSTM RNN models for speech recognition," *Proc. of ICASSP*, pp. 5020–5024, 2016.

[15] Z. Huang, J. Li, S. M. Siniscalchi, I. Chen,J. Wu, "Rapid Adaptation for Deep Neural Networks through Multi-Task Learning," *Proc. of Interspeech*, pp. 3625-3629, 2015.

[16] Y. Zhao, J. Li, J. Xue, Y. Gong, "Investigating online low-footprint speaker adaptation using generalized linear regression and click-through data," *Proc. of ICASSP*, pp. 4310–4314, 2015.

[17] T. Tan, Y. Qian, M. Yin, Y. Zhuang, K. Yu, "Cluster adaptive training for deep neural network," *Proc. of ICASSP*, pp. 4335-4329, 2015.

[18] 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.

[19] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *Proc. of ICLR*, 2015.

[20] J. K. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," *Advances in Neural Information Processing Systems*, pp. 577–585, 2015.