Multi-stream CNN Based Accented English Automatic Speech Recognition System

Li Tao 31520211154017[1], **Li Xiang 31520211154009**[1],
Wang Feng 31520211154019[1], **Hu Yuwen 36620211150339**[1]
[1] Xiamen University
31520211154017@stu.xmu.edu.cn, 31520211154009@stu.xmu.edu.cn, 31520211154019@stu.xmu.edu.cn,
36620211150339@stu.xmu.edu.cn,

Abstract

In this paper, we present multi-stream CNN based accented english automatic speech recognition system. Automatic Speech Recognition (ASR) task where the non-native English speakers have various accents reduces the accuracy of ASR system. In order to solve this problem, we have made a lot of attempts. For DNN-HMM hybrid system, we tried several acoustic models (AM) and chose Multistream CNN. Then we explored various speaker/accent embeddings, for example, i-vector or x-vector, to further improve the accuracy of accented ASR systems. Experiments showed that using embeddings which capture the accent/speaker-relevant information as auxiliary inputs can significantly improve the accuracy of accented ASR system. Finally, we trained a TDNN-LSTM language model for lattice rescoring to get better results. Compared with our baseline system, we achieved relative word error rate (WER) improvements of 40.6% and 35.6% on the development set and evaluation set respectively.

Introduction

The standard English ASR system has been able to obtain a high recognition accuracy rate and meet the commercial requirements of certain scenarios. However, numerous scientific research suggest that accent effects the recognition rate of ASR system in a large extent. Due to the inconsistency of the accent itself, the variability of speech speed and phoneme pronunciation, and the scarcity of accented speech data, the accented English recognition is still a challenging subject.

The traditional acoustic model employed the parameter-adaptive GMM-HMM model which is based on statistical learning methods, but the low efficiency rate and poor robustness limit its further application. With the rise of deep neural network(DNNs), it is found that DNNs have stronger representation and modeling capabilities than GMM. In (Hinton et al. 2012), a contextual DNN-HMM hybrid system was proposed and then applied on five large-vocabulary continuous speech recognition tasks, there was an average relative WER improvement of 19% compared with GMM-HMM. (Povey et al. 2018a) proposed two innovative improvements on the basis of TDNN[10], thereby further developed the context-sensitive DNN-HMM hybrid system. In (Povey et al. 2018a), A method of matrix called Singular Value Decomposition (SVD) is used to compress the parameters of model, then the orthogonal matrix is generalized to the semi-orthogonal matrix for the non-square matrix in order to maintain the modeling power. Meanwhile, a great deal of research work have been invested in improving the robustness of the system. The multistream CNN (Han et al. 2020), inspired by the multistream self-attention architecture (Han et al. 2019) but without the multi-headed self-attention layers, processes the input speech frames in multiple streams. Each stream stacks TDNN-F (Povey et al. 2018a) with specific dilation rate for diversity. It is reported that a system which combined multistream CNN with self-attentive SRU[7] has achived a SOTA Speech Recognition on Librispeech. In addition, SpecAugment data augmentation method (Park et al. 2019), which gives masks on the spectrogram of input utterances, are performed during training. We have thoroughly investigated the recent acoustic models that work well and compared their performance on Accented English ASR. We found that multistream CNN has the best effect, but the improvement relative to the baseline is still not very obvious. Therefore, we consider adding some representative information to the input of the network.

In the field of speaker recognition, one popular approach to address speaker variability is to augment the DNN's input features with auxiliary features which embed speaker information. I-vector (Dehak et al. 2010) which captures both speaker and environment specific information has been shown to be effective for ASR task (Peddinti et al. 2015), and DNN-based embeddings such as x-vector (Snyder et al. 2018) have replaced i-vector in many certain circumstances. X-vector framework extracts fixed-dimension speaker embeddings from variable-length utterances, has achieved superior performance especially when given sufficient training data compared with i-vector framework. For accented speech, the information of tone and speaking habit is of great importance. In addition, considering the scarcity of both accent-related corpus and speaker-related corpus data, we can treat an accent as a speaker. There has been some literature on accent and dialect adaptive speech recognition. In (DeMarco and Cox 2013), i-vector has been used to characterize different native British accents. Accent-dependent i-vector and x-vector are used in (Chen et al. 2015; Turan, Vincent, and Jouvet 2020) to improve the performance of NN-based multi-accent speech recognition, respectively. we extracted four types of representations (i.e., spk-ivector, accent-ivector, spk-xvector, accent-xvector) and compared their efficiency in accented ASR system. Spk-ivector is a regular i-vector while accent-ivector use the same strategy as spk-ivector, but classify the accents in the training set. Similar to x-vector, accent-xvector system use TDNNs with a pooling layer that collect statistics over time and the segment level representation is then used as the input of two fully connected layers to predict the accent labels rather speaker labels. After training, the accent embedding is extracted from the affine component of the first fully connected layer. We found that both speaker and accent relevant embeddings can improve the effect of ASR. What's more, x-vector embedding performs better than i-vector embedding.

Finally, under the rescoring of language model, our best system achieved a relative WER improvements of 40.6% on development set and 35.6% on evaluation set compared with the baseline.

The rest of this paper is structured as follows. In Section 2, we describe the details of our systems. In Section 3, we provide the experimental setups and discuss the results from various approaches. Finally, we conclude our work in Section 4.

System Structure

Acoustic Modeling

We followed the conventional steps to train hybrid GMM-HMM acoustic models referring to Kaldi (Povey et al. 2011) recipe for CHIME6[1]. It has been shown that a sequence-level training criteria like lattice-free maximum mutual information (LF-MMI) performs better than frame-level criteria for ASR (Povey et al. 2016). Our systems are based on the TDNN-F (Povey et al. 2018a) acoustic model using LF-MMI training criterion. We experimented various network structures. All of our experiments are based on Kaldi toolkit.

- **TDNN-F**: TDNN-F model is the first 11-layers of TDNN-F in the recipe for CHIME6 of the Kaldi (`egs/chime6/s5_track1/local/chain/tuning/run_tdnn_1b.sh`).
- **CNN-TDNNF-Attention**: The CNN-TDNNF-Attention model consists of 1 CNN layer followed by 11 time-delay layers and a time-restricted self-attention layer (Povey et al. 2018b), and apply SpecAugment (Park et al. 2019) layer on top of the architecture thus enable it more robust. The CNN layer has a kernel size of 3x3 and a filter size of 64. The 11-layers TDNN-F share the same configuration as the previous illustrated TDNN-F except substitutes the first TDNN layer with a TDNN-F layer, which has 1536 nodes, 256 bottleneck nodes and no time stride. The attention block has 8 heads, the value-dim and key-dim are set to 128 and 64 respectively, the context-width is 10 with the same number of left and right inputs, and time stride is 3.
- **Multistream CNN**: Position 5-layers CNN to better accommodate the top SpecAugment layer, followed by 11-layers multistream CNN (Han et al. 2020).

Multistream CNN Architecture

Multistream CNNs have shown its superiority in robust speech recognition, for its diversity in temporal resolutions across multiple parallel streams would achieve stronger robustness.

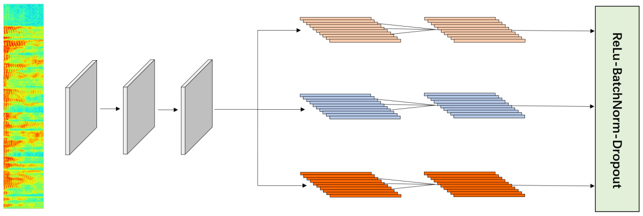

Figure 1: Schematic diagram of the multistream CNN acoustic model architecture.

As shown in Figure 1, the input speech frames is first processed by a few initial single stream CNN layers, which can be TDNN-F or 2D-CNN, and then enters multiple specific branches that stacked by TDNN-F layers. To achieve the diversity of temporal resolution, every branched stream has a unique dilation rate which corresponds to the time stride in the TDNNs. Each dilation rate is chosen from the default subsampling rate (3 frames) in order to make TDNN-Fs better streamlined with the training and decoding process when given input speech frames are subsampled.

In our multistream CNN network, the log-mel spectrogram is first randomely masked in both frequency and time by a SpecAugment layer, then 5 layers of 2D-CNN are positioned to better accommodate the features. We use 3x3 kernels for the 2D-CNN layers, the filter size of the first two layers is 64, the third and the fourth is 128, and the last is 256. Every other 2D-CNN layer we apply frequency band subsampling with a rate of 2. In the multi-stream part, each branch we stacked 11 layers TDNN-F with 512 nodes and 128 bottleneck nodes. We employ 3 streams with the 6-9-12 dilation rate configuration which means TDNN-Fs of each streams have 6,9,12 time-stride respectively. The output embeddings of multiple streams are then concatenated, and followed by ReLu, batch normalization and a dropout layer.

Accent/speaker Embeddings

I-vector is a popular technology in the field of speaker recognition. It was motivated by the success of the Joint Factor Analysis (JFA) (Kenny et al. 2007). JFA is used to construct the subspace of speaker and channel separately, proposes powerful tools to model the inter-speaker variability and to compensate for channel/session variability in the context of GMM. However, (Dehak 2009) proved that channel factors estimated using JFA, which are supposed to model only channel effects, also contain information about speakers. Thus, i-vector methods construct a low-dimensional subspace, termed the total variability space. This space contains

[1] https://github.com/kaldi-asr/kaldi/tree/master/egs/chime6/s5_track1

factors of both speaker and channel variability. In this way, i-vector models both speaker and channel information, and characterizes most of the useful speaker-specific information in a fixed and low dimensional feature.

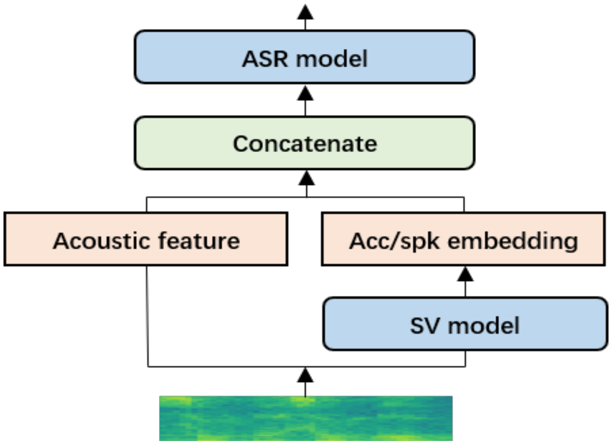

Figure 2: Embeddings extraction and integration with ASR model.

Due to the powerful representation capabilities of DNNs, x-vector is now a powerful representation for speaker recognition. On one hand, the TDNN-based architecture models the short-term context; on the other hand, the statistics pooling layers process the whole information across the time dimension so that subsequent layers operate on the entire segment. The DNN architecture used to extract x-vectors is outlined in Table 1. The splicing parameters for the five TDNN layers are :{t - 4,t - 3,t - 2,t - 1,t,t + 1,t + 2,t + 3,t + 4}, {t - 2,t,t + 2}, {t - 3,t,t + 3}, {t}, {t}. The statistics pooling layer is used to calculate the mean and standard deviation on all frames of the input segment, and followed by upper layers at the segment level with a softmax output layer. Segment-level embeddings are extracted from the 512-dimensional affine component of the first fully connected layer. Speaker embeddings which capture both speaker and environment specific information have been shown to be useful for ASR task. For accented speech, the information of tone and speaking habit of each accent is specially important. We have explored four types of representations as auxiliary inputs in a neural network to further improve the accuracy of accented ASR system, the procedure is shown in Figure 2. In the procedure, we first extract embeddings of each utterance, and then concatenate them to each frame of the utterance as a compensatory input for neural network.

Neural-Network Alignment

The phonetic alignment generated by GMM could be inaccurate, so we trained a DNN model using the frame-level criteria to get a better alignment, which substitutes the GMM alignment to train acoustic models with LF-MMI training criterion.

Layer	Layer Type	Context	Size
Frame1	TDNN	{t-4:t+4}	512
Frame2	TDNN	{t - 2,t,t + 2}	512
Frame3	TDNN	{t - 3,t,t + 3}	512
Frame4	TDNN	{t}	512
Frame5	TDNN	{t}	1500
Stat pool		{0,T}	2 x 1500
Segment6	Affine	{0}	512
Segment7	Affine	{0}	512
Softmax		{0}	Num.accent/spk

Table 1: Accent/spk-xvector architecture.

System	Features	WERs (%)	on eval
TDNN-F (baseline)	MFCC	9.12	9.31
TDNN-F	MFCC+Pitch	8.97	9.18
CNN-TDNNF-Attention	MFCC+Pitch	8.92	9.12
Multistream CNN	MFCC+Pitch	**8.86**	**9.08**

Table 2: Effect of different acoustic models.

Language Model Rescoring

During decoding, a 4-gram language model (LM) was used to generate the lattice and score. This model has the problem of data sparsity, because we only used the transcription of the training data to train it. To get better results, we trained a 4-layer TDNN-LSTM LM for lattice rescoring. It's worth mentioning that N was set to 20 for n-best rescoring. And when training TDNN-LSTM LM we also use Librispeech (Panayotov et al. 2015) text in addition to the transcription of the training set.

Experimental Results

Data Sets and Augmentation

Our experiments were conducted on the accented English data sets (16kHz) of eight countries provided by Datatang[2], including American (US), British (UK), Chinese (CHN), Korean (KR), Japanese (JPN), Russian (RU), Portuguese (PT) and Indian (IND). Each accented speech data is collected from 40-110 speakers and recorded by Android devices or iPhones in quiet house acoustic environment. The speakers are gender balanced, age 20 to 60. Every accented data has about 20 hours. The speech content consist of daily communication and interaction with smart devices. Training set, development set, and evaluation set are about 148 hours, 14 hours, and 21 hours respectively. In the evaluation set, some accent data not included in the training set have also been added, these data that have not seen will be used to test the generalization of the system.

We have augmented the training data by changing the speed of the audio signal, producing 3 versions of the original signal with speed factors of 0.9,1.0 and 1.1 (Ko et al. 2017), and then apply volume perturbation. All the systems share the same type of data augmentation techniques. In addition, SpecAugment which gives masks on the spectrogram of input utterances, is performed during training.

[2] https://www.datatang.com

Embeddings	Dev	Eval
[M1] w/o embeddings	8.86	9.08
[M2] Spk-ivectors	7.18	8.01
[M3] Accent-ivectors	7.17	7.95
[M4] Accent-ivectors + spk-ivectors	7.02	8.02
[M5] Spk-xvectors	7.04	7.76
[M6] Accent-xvectors	7.02	7.89
[M7] Accent-xvectors + spk-xvectors	**6.95**	**7.74**

Table 3: WERs (%) achieved by multistream CNN with various input embeddings.

Effect of Acoustic Model

Firstly, we choose the pure 11-layers TDNN-F to implement a baseline system. The features are 40-dimensional high resolution MFCC computed with a 25ms window and shifted every 10ms. Table 2 shows that Kaldi pitch features can give improvements on tonal languages for ASR systems. So we choose the 43-dimensional MFCC with pitch as acoustic features for our systems.

Secondly, we replace the network with several models as illustrated in Sec. 2.1. The results of different network architectures are shown in Table 2. Finally, the Multistream CNN is trained and get the best performance. As a result, we yield relative WER improvements of 2.8% and 2.4% on the development set and evaluation set respectively compared with the baseline system.

Effect of Accent/speaker Embeddings

To further improve performance of the ASR system, we have explored various embeddings as auxiliary features while share the same acoustic model and training strategy. The WERs achieved by the 7 systems are reported in Table 4.

Model M1 is a model trained without any auxiliary embeddings, and its WER is highest in Table 4, so we can conclude that using embeddings as complementary features can significantly improve the performance of accented English speech recognition system.

We observe that both spk-ivectors model (M2) and accent-ivectors model (M3) can achieve the similar WER reduction, and the spk-ivectors + accent-ivectors model (M4) can get a further improvement on development set, which means that both speaker-relevant information and accent-relevant information are helpful for accent adaption in accented ASR. This results can also be observed from M5 to M7, which are the models applied the x-vector embeddings as auxiliary features.

In Table 4, we observe that the x-vector embeddings(M5, M6, and M7) outperform i-vector embeddings(M2, M3, M4). Compared with Model M4, which utilize the combination of spk-ivectors and accent-ivectors, Model M6 which just augment input with accent-xvectors get the same 7.02% WER on development set. This is because the x-vector has a more powerful capability to characterize accent-relevant information. When combining the accent-xvectors and spk-xvectors in Model M7, we achieve a relative WER reduction of 21.5% on development set, and 14.7% on evaluation set, compared with baseline model M1.

Embeddings	Dev	Eval
w/o embeddings	8.86	9.08
Spk-ivectors	7.18	8.01
Accent-ivectors	7.17	7.95
+ spk-ivectors	7.02	8.02
Spk-xvectors	7.04	7.76
Accent-xvectors	7.02	7.89
+ spk-xvectors	**6.95**	**7.74**
+ LM rescoring	**5.41**	**5.99**

Table 4: WERs (%) achieved by multistream CNN with various input embeddings.

System	Features	Embeddings	**WER** (%) **on dev**	**WER** (%) **on eval**
Baseline (TDNN-F)	MFCC	-	9.12	9.31
Multistream CNN	MFCC+Pitch	accent-xvectors +spk-xvectors	6.95	7.74
Multistream CNN +LM rescoring	MFCC+Pitch	accent-xvectors +spk-xvectors	**5.41**	**5.99**

Table 5: Effect of language model rescoring.

Effect of Language Model Rescoring

We selected the best model M7 from Table 4, and applied N-best rescoring using the TDNN-LSTM LM. As shown in Table 5, the best system is taken to be the one that performs best on the development set. The input feature is 43-dimensional Kaldi MFCC with Pitch appended with spk-xvectors and accent-xvectors, acoustic model is Multistream CNN, and with LM rescoring we got 5.41% WER on development set and 5.99% on evaluation set. Compared with baseline system, we achieve relative WER improvements of 40.6% and 35.6% on the development set and evaluation set respectively.

Conclusions

This paper presents the systems for the Accented English ASR. We explored various approaches to improve the accuracy of accented ASR system as following aspects. The multistream CNN which parallel process input with different time strides gives the acoustic model stronger robustness. In addition, accent/speaker embeddings can characterize inner accent-relevant information, and further bring improvement in accented ASR. Moreover, results show that x-vector embeddings outperform i-vector embeddings. Finally, a language model (LM) was trained for lattice rescoring. We chose the best model from previous attempts to do LM rescoring and achieved a great improvement in comparison with our baseline.

References

Chen, M.; Yang, Z.; Liang, J.; Li, Y.; and Liu, W. 2015. Improving deep neural networks based multi-accent Mandarin speech recognition using i-vectors and accent-specific top layer. In *INTERSPEECH*, 3620–3624.

Dehak, N. 2009. *Discriminative and generative approaches for long-and short-term speaker characteristics modeling: application to speaker verification*. Ph.D. thesis, École de technologie supérieure.

Dehak, N.; Kenny, P. J.; Dehak, R.; Dumouchel, P.; and Ouellet, P. 2010. Front-end factor analysis for speaker verification. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(4): 788–798.

DeMarco, A.; and Cox, S. J. 2013. Native accent classification via i-vectors and speaker compensation fusion. In *INTERSPEECH*, 1472–1476.

Han, K. J.; Pan, J.; Tadala, V. K. N.; Ma, T.; and Povey, D. 2020. Multistream CNN for Robust Acoustic Modeling. *arXiv: Audio and Speech Processing*.

Han, K. J.; Prieto, R.; Wu, K.; and Ma, T. 2019. State-of-the-Art Speech Recognition Using Multi-Stream Self-Attention With Dilated 1D Convolutions. arXiv:1910.00716.

Hinton, G.; Deng, L.; Yu, D.; Dahl, G.; rahman Mohamed, A.; Jaitly, N.; Senior, A.; Vanhoucke, V.; Nguyen, P.; Sainath, T.; and Kingsbury, B. 2012. Deep Neural Networks for Acoustic Modeling in Speech Recognition. volume 29, 82.

Kenny, P.; Boulianne, G.; Ouellet, P.; and Dumouchel, P. 2007. Joint factor analysis versus eigenchannels in speaker recognition. *IEEE Transactions on Audio, Speech, and Language Processing*, 15(4): 1435–1447.

Ko, T.; Peddinti, V.; Povey, D.; Seltzer, M. L.; and Khudanpur, S. 2017. A study on data augmentation of reverberant speech for robust speech recognition. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 5220–5224.

Panayotov, V.; Chen, G.; Povey, D.; and Khudanpur, S. 2015. Librispeech: An ASR corpus based on public domain audio books. In *ICASSP 2015 - 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.

Park, D. S.; Chan, W.; Zhang, Y.; Chiu, C.-C.; Zoph, B.; Cubuk, E. D.; and Le, Q. V. 2019. SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition. In *Interspeech 2019*, 2613–2617.

Peddinti, V.; Chen, G.; Povey, D.; and Khudanpur, S. 2015. Reverberation robust acoustic modeling using i-vectors with time delay neural networks. In *INTERSPEECH*, 2440–2444.

Povey, D.; Cheng, G.; Wang, Y.; Li, K.; Xu, H.; Yarmohammadi, M.; and Khudanpur, S. 2018a. Semi-Orthogonal Low-Rank Matrix Factorization for Deep Neural Networks. In *Interspeech 2018*, 3743–3747.

Povey, D.; Ghoshal, A.; Boulianne, G.; Burget, L.; Glembek, O.; Goel, N.; Hannemann, M.; Motlicek, P.; Qian, Y.; Schwarz, P.; Silovsky, J.; Stemmer, G.; and Vesely, K. 2011. The Kaldi Speech Recognition Toolkit. In *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*.

Povey, D.; Hadian, H.; Ghahremani, P.; Li, K.; and Khudanpur, S. 2018b. A Time-Restricted Self-Attention Layer for ASR. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 5874–5878.

Povey, D.; Peddinti, V.; Galvez, D.; Ghahremani, P.; Manohar, V.; Na, X.; Wang, Y.; and Khudanpur, S. 2016. Purely Sequence-Trained Neural Networks for ASR Based on Lattice-Free MMI. In *Interspeech 2016*, 2751–2755.

Snyder, D.; Garcia-Romero, D.; Sell, G.; Povey, D.; and Khudanpur, S. 2018. X-Vectors: Robust DNN Embeddings for Speaker Recognition. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 5329–5333.

Turan, M. A. T.; Vincent, E.; and Jouvet, D. 2020. Achieving multi-accent ASR via unsupervised acoustic model adaptation. In *INTERSPEECH 2020*.