The THUEE Systems for the NIST Open Speech Analytic Technologies 2019 Evaluation

Guan-Bo Wang, Qian Liu, Zeyu Zhao, Zhiqiang Lv, Wei-Qiang Zhang
Department of Electronic Engineering, Tsinghua University, Beijing 100084, China
wqzhang@tsinghua.edu.cn

***Abstract*—This paper introduces the systems of THUEE for the NIST Open Speech Analytic Technologies 2019 Evaluation (OpenSAT19). We first describe the speech activity detection (SAD) system, which combines a convolutional recurrent neural network (CRNN) and a recurrent neural network (RNN). In order to improve the performance of our system in low signal-noise ratio conditions, we also add a speech-enhancement module, a one-dimensional dilation-erosion module, and a model ensemble module, all of which contribute significantly. Next, we introduce the automatic speech recognition (ASR) system. We utilize speech augmentation and multilingual bottleneck feature strategies. We also build several acoustic models, such as TDNN-F, TDNN, LSTM and BLSTM. Last, we describe the keyword search (KWS) system. We build several KWS subsystems based on ASR subsystems, and use proxy word method to deal with out-of-vocabulary (OOV) words.**

I. Notable Highlights

In our SAD system, we combine a convolutional recurrent neural network (CRNN) and a recurrent neural network (RNN) to make our system more robust. In addition, we add a speech-enhancement module and a one-dimensional dilation-erosion module to our SAD system.

In our ASR system, the main highlight is the diversity of our acoustic models and data augmentation of language models. While training half of the acoustic models, we use MFCC and pitch features as input acoustic features, and use i-vectors for speaker adaptation. As for the other half acoustic models, we use multilingual bottleneck features for model training. As for acoustic models, TDNN-F, TDNN-LSTM, TDNN-BLSTM are used to enhance the performance of the combined system. Besides, we find some publicly available Pashto text and select the text similar to training transcripts to improve the performance of language models.

Is our KWS system, different parameters of the ASR systems are used to expand the diversity. Besides, we apply a data augmentation technique that we re-segment the audio with a basic ASR system to make the transcription more precise.

II. Data Resource

For the SAD system, we use data resources as follows:

Training:

- LDC2017E13_OpenSAT_2017_VAST_Dev_Data_V2
- LDC2019E36_SAFE-T_Speech_Recording_Training_Data_Transcripts_V1
- LDC2019E37_SAFE-T_Corpus_Speech_Recording_Audio_Training_Data_R1
- IARPA_BABEL_BP_104

Development:

- LDC2019E38_SAFE-T_Speech_Recording_Development_Audio_and_Transcripts_V1.1
- LDC2019E49_2019_OpenSAT_Video_Annotation_for_Speech_Technology_Development_Data_V1.0
- LDC2017E16_OpenSAT_2017_IARPA_Babel_Eval_Data

Evaluation:

- LDC2019E50_2019_OpenSAT_Public_Safety_Communications_Simulation_Evaluation_Data
- LDC2019E54_2019_OpenSAT_Video_Annotation_for_4Speech_Technology_Evaluation_Data
- LDC2017E16_OpenSAT_2017_IARPA_Babel_Eval_Data

For the ASR and KWS systems, we use data resources as follows:

Training:

- IARPA_BABEL_BP_104

Development and evaluation:

- LDC2017E16_OpenSAT_2017_IARPA_Babel_Eval_Data

III. Algorithmic Description/Approach

A. SAD

1) System Overview: Our proposed system is shown in Fig. 1. Firstly, input audios are enhanced by speech-enhancement module. Next, we extract fbank features of raw audios and enhanced audios, and then put the features into both a convolutional recurrent neural network (CRNN) and a recurrent neural network (RNN). The two networks will output speech existence in every frame separately. Finally, we pass the output to the model ensemble module and one-dimensional dilation-erosion module and get the final output.

2) Feature Extraction: In this part of the system, we follow the work in [1]. To imitate non-linear response to the sound spectrum of human ear, we choose the log Mel-scale Filter Bank energies (fbank) feature. First, we convert sampling rate of input audio to 16 kHz, and divide each audio into frames. The frame length is 20 ms and the frame shift is 10 ms. Then we extract fbank feature of each frame, applying 40 mel-scale filters on the magnitude spectrum, which cover the entire range from 0 to 8000 Hz. After that, we take logarithm on the amplitude and then get the fbank feature. Finally, in order to be

The corresponding author is Wei-Qiang Zhang.
This work was supported by the National Natural Science Foundation of China under Grant No. U1836219.

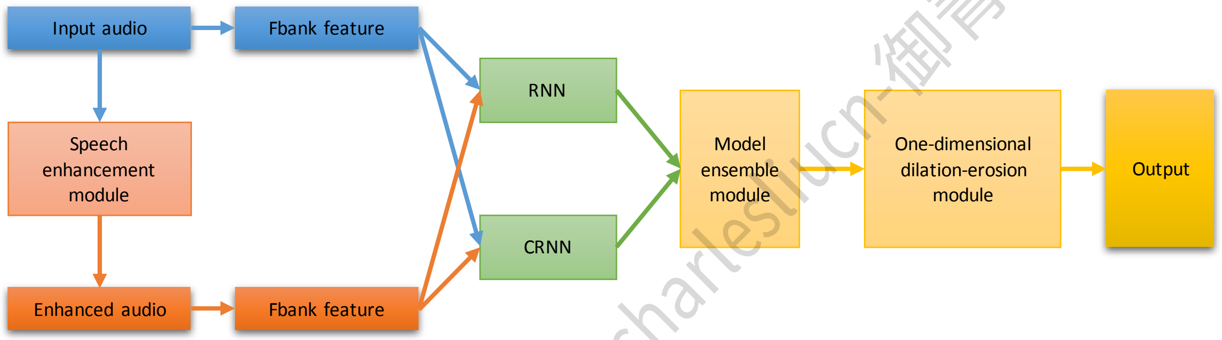

Fig. 1. Overall architecture of SAD system.

prepared to be fed into neural networks, the extracted feature is normalized to zero mean and unit standard deviation.

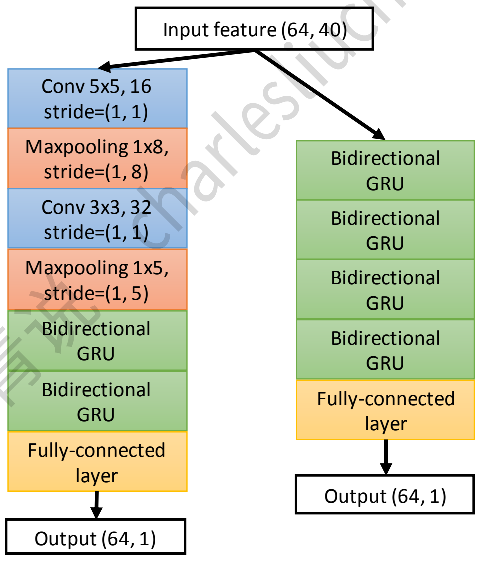

Fig. 2. The architecture of CRNN and RNN of SAD system.

3) Neural Network: In this part of our proposed system, we train two different neural networks, a CRNN and an RNN. Both networks are shown in Fig.2. We cut input feature into 64-frame length segments, so the input shape is 64×40. In CRNN structure, the first two layers are 2-dimensional convolution (Conv-2D) layers, and the next two layers are bi-directional gated recurrent unit (bi-GRU) layers. The final layer is a fully-connected layer, which outputs whether speech exists in each frame. In RNN structure, there are 4 bi-GRU layers after the input layer, and the final layer is a fully-connected layer, which outputs whether speech exists in each frame.

4) Speech-Enhancement: In this module, our work is based on a regression approach to speech enhancement based on deep neural networks (DNN) [2]. To train the speech-enhancement model, we utilize the same datasets, the same neural network, and the same training method mentioned or provided in [2]. We add noise audio to clean speech audio and get noisy speech. Then both noisy speech and clean speech are fed into the DNN as input and output of the network separately. In this way, we get a DNN based speech-enhancement module. We also listen to the output of the module, and although the speech enhancement results don't sound perfect, it does improve the SAD performance.

5) One-Dimensional Dilation-Erosion: Due to the defect of speech-enhancement, speech is inevitably weakened when background noise is reduced. Therefore, some "holes" that appear in speech segments in output need to be patched, and some boundaries need to be expanded. The dilation-erosion algorithm is widely used in computer vision area, which means dilating or eroding white area on binary image. In our system output, each audio is divided into several segments marked speech or nonspeech, and this is just like a one-dimensional binary image. Therefore, our one-dimensional dilation-erosion algorithm is dilating or eroding speech segments of output.

6) Model Ensemble: In order to get better performance, we add model ensemble module to our proposed SAD system, a commonly applied strategy. We choose the BUT phoneme recognizer [3] as our ensemble model, which is also our baseline system. The method for model ensemble is WCombSum algorithm [4] used in keyword search (KWS) area. WCombSum is computed as follows:

$$s = \sum_{i=1}^{N} w_i \times s_i \tag{1}$$

where w_i denotes weight of each system, and s_i denotes confidence score for each system. We normalize the output of each system into a range of 0 to 1, so that we can find the optimal threshold θ within a certain range to minimize the $DCF(\theta)$.

B. ASR

1) Speech Augmentation: In order to enhance the robustness of our ASR systems, we adopt speed perturbation as well as volume perturbation methods to the original training speech audio. For each audio file in the training data, we use both 1.1x and 0.9x speed perturbation. Afterwards, volume perturbation is conducted for all the audio. These perturbation steps make our system more invariant to speed and volume of the evaluation data.

2) Acoustic Features: The acoustic models of our systems are based on two kinds of acoustic features. One is high-resolution MFCC feature with pitch feature, and the other is multilingual TDNN-based bottleneck feature. For MFCC-based acoustic models, we also use i-vector feature for speaker adaptation. For multilingual bottleneck features, we first train GMM-HMM acoustic models using MFCC and pitch features for 24 kinds of BABEL languages to obtain state-level labels. Then we extract bottleneck features using 6-layer TDNN. These two kinds of acoustic features are used to train our acoustic models with the same neural network structure separately, in order to improve the performance for system combination.

3) Speech Re-Segmentation: In BABEL ASR evaluation, we are provided with the segment-level transcription. However, the quality of the transcripts may not be good enough for acoustic model training. Therefore, we decode these audio with a triphone acoustic model and a biased language model, and then select the audio that matches the given transcripts as cleaned training data.

4) Lexicon Setup: In our system, we simply use the original lexicon provided by the Pashto language pack. It is worth noting that not all the words in the lexicon appear in the training transcripts, but we keep all the words in order to maximize our vocabulary size to reduce OOV rate on the evaluation data.

5) Acoustic Modeling: We try to use several kinds of acoustic models in this low-resource task for later system combination. Each acoustic model corresponds to a sub-system. We mainly use state-of-the-art TDNN-F models [5] with different set of parameters. The difference between these TDNN-F models includes the depth of the neural networks, the number of hidden units in each layer and so on.

Besides, we also use other models like TDNN [6], TDNN-LSTM, TDNN-BLSTM, TDNN with attention layer as auxiliary acoustic models. In addition to TDNN-based models, traditional LSTM and Bidirectional LSTM models are also trained.

6) Speaker Adaptation: I-vectors are fixed-length vectors containing speakers information, and have become a common technique for speaker recognition. In our TDNN-based sub-systems, we train the i-vectors based on a diagonal UBM for speaker adaptation [7].

7) Language Modeling: For language modeling in our sub-systems, an interpolated 4-gram model is used. One language model is trained with transcripts of the training data. The other 4-gram model is trained with selected out-of-domain text data.

The out-of-domain Pashto text is mainly web text obtained from [8]. In order to make the best of it, we first remove all the punctuation marks and other meaningless symbols. Then we want to select the text similar to the training transcripts. This data selection step can be considered as a task of text similarity analysis. An open source toolkit, XenC [9], is used for the data selection based on the difference of cross entropy. The 4-gram model is then linearly interpolated with original language model trained from training transcripts. As for the interpolation parameter, it is determined to minimize the perplexity on development data.

8) Decoding: In our systems, we use a WFST-based method for decoding based on Kaldi Toolkit [10]. For the first-pass decoding, we simply use the interpolated n-gram model as the decoding language model. For a part of sub-systems with TDNN-based acoustic models, a two-layer LSTM language model is trained for lattice rescoring [11]. The training data of the LSTM language model is all the available text, including in-domain and out-of-domain text.

9) System Combination: In order to enhance the diversity of our sub-systems, we use a lot of different setup. There are 18 sub-systems with different acoustic features, acoustic models and decoding setup. After we obtain the recognition result of each sub-system, we adopt ROVER method [12] to combine all the outputs.

C. KWS

1) System Overview: Our KWS systems are based on ASR systems, some of which are used for this competition but some of which are not, using the multi-language bottle-neck features. After training the ASR systems, we decode the speech to get word-lattice. Then, we perform the keyword search process on the word-lattice. Particularly, we train a grapheme-to-phoneme (G2P) model to deal with the OOV words, which, in fact, generates hundreds of proxy word for each OOV word [13], although for some extremely short words the G2P method can not work on them. We will discuss the KWS systems in details as bellow.

2) Decode Setting for KWS: Unlike an ASR system, for which we need to maximize the posterior probability of the 1-best result, a KWS system should output a word-lattice containing much more decoding results. In other word, for an ASR system, the posterior probability density function of the decoding results usually has one peak and the shape of the posterior probability density function are sharper than that of a KWS system. To accomplish this purpose, we set a bigger beam width when decoding the speech, and, compared to the systems with small beam width used in ASR system, we get better KWS performance.

3) Out of Vocabulary: To deal with the OOV words, we train a G2P model based on an HMM-GMM (Hidden Markov Model, HMM, Gaussian Mixture Model, GMM) model to get hundreds of proxy words for each OOV word. However, for some very short word, the G2P model cannot generate proxy word for them properly. Even though we may force the G2P model to do this work, there will be a lot of false alarms in the keyword search result. In short, we cannot deal with the very-short word well.

4) System Combination: According to the performance of each KWS system on the development set, we set each KWS system a weight to combine them using the KWS combination tools in Kaldi.

IV. Results on the DEV set

A. SAD

The final performance in DEV set of our proposed systems is shown in Table I.

TABLE I
PERFORMANCE OF THE SAD SYSTEM ON DEV SET

dataset	P_{FN}	P_{FP}	DCF
PSC_dev	0.0602	0.0427	0.0558
VAST_dev	0.0517	0.1669	0.0805
Babel_dev	0.0385	0.1257	0.0603

B. ASR

The WER of each sub-system on the DEV set is shown in Table II. The best individual system is based on TDNN-F acoustic model. After system combination using ROVER method, the system achieves a WER of 43.3% on the development set.

TABLE II
WER OF THE ASR SYSTEM ON THE DEV SET

Acoustic Model	WER on the DEV set
TDNN-F	46.9%
TDNN	52.4%
TDNN-LSTM	48.2%
TDNN-BLSTM	51.0%
TDNN-Attend	52.1%
LSTM	52.1%
BLSTM	50.9%
ROVER	43.3%

C. KWS

The results are shown in Table III.

V. Hardware Description

The hardware of our proposed system is shown in Table IV.

TABLE III
ATWV AND MTWV OF THE KWS SYSTEM ON THE DEV SET

System	ATWV	MTWV
tdnnf_7q	0.5024	0.5222
tdnnf_7q_17	0.5011	0.5245
tdnnf_7q_19	0.5054	0.5243
tdnnf_7q_clean	0.4981	0.5070
tdnnf_7q_17_clean	0.4983	0.5088
tdnnf_7q_19_clean	0.4981	0.5148
tdnn_attend	0.4144	0.4235
tdnn_basic	0.4332	0.4451
tdnn_lstm	0.3906	0.3943
tdnn_blstm	0.4383	0.4429
tdnn_lstm_swbd	0.4703	0.4826
tri5_dnn_smbr	0.4276	0.4406
tri5_bresgru	0.4467	0.4531
tri5_lru	0.3997	0.4083
tri5_bresdgru	0.4419	0.4508
combined	**0.5030**	**0.5664**

TABLE IV
HARDWARE DESCRIPTION

OS	CentOS 7.4 64-bit
CPU num	2
CPU description	28,Intel(R) Xeon(R) CPU E5-2680 v4 @ 2.40GHz 112,Intel(R) Xeon(R) CPU E5-4650 v4 @ 2.20GHz
GPU num	1
GPU description	Tesla P100 SMX2 16GB
RAM	256GB
RAM per CPU	128GB
Used Disk Storage	About 1TB

References

[1] Y.-H. Shen, K.-X. He, and W.-Q. Zhang, "Learning how to listen: A temporal-frequential attention model for sound event detection," in *Interspeech*, 2019.

[2] Y. Xu, J. Du, L.-R. Dai, and C.-H. Lee, "A regression approach to speech enhancement based on deep neural networks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 1, pp. 7–19, 2014.

[3] P. Ace, P. Schwarz, and V. Ace, "Phoneme recognition based on long temporal context," Ph.D. dissertation, Citeseer, 2009.

[4] J. Mamou, J. Cui, X. Cui, M. J. Gales, B. Kingsbury, K. Knill, L. Mangu, D. Nolden, M. Picheny, B. Ramabhadran *et al.*, "System combination and score normalization for spoken term detection," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2013, pp. 8272–8276.

[5] D. Povey, G. Cheng, Y. Wang, K. Li, H. Xu, M. Yarmohammadi, and S. Khudanpur, "Semi-orthogonal low-rank matrix factorization for deep neural networks." in *Interspeech*, 2018, pp. 3743–3747.

[6] V. Peddinti, D. Povey, and S. Khudanpur, "A time delay neural network architecture for efficient modeling of long temporal contexts," in *Interspeech*, 2015.

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

[8] D. Goldhahn, T. Eckart, and U. Quasthoff, "Building large monolingual dictionaries at the leipzig corpora collection: From 100 to 200 languages." in *LREC*, vol. 29, 2012, pp. 31–43.

[9] A. Rousseau, "Xenc: An open-source tool for data selection in natural language processing," *The Prague Bulletin of Mathematical Linguistics*, vol. 100, pp. 73–82, 2013.

[10] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz *et al.*, "The Kaldi speech recognition toolkit," in *2011 IEEE Workshop on Automatic Speech Recognition and Understanding*, 2011.

[11] X. Liu, Y. Wang, X. Chen, M. J. Gales, and P. C. Woodland, "Efficient lattice rescoring using recurrent neural network language models," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2014, pp. 4908–4912.

[12] J. G. Fiscus, "A post-processing system to yield reduced word error rates: Recognizer output voting error reduction (ROVER)," in *1997 IEEE Workshop on Automatic Speech Recognition and Understanding Proceedings*, 1997, pp. 347–354.

[13] G. Chen, O. Yilmaz, J. Trmal, D. Povey, and S. Khudanpur, "Using proxies for oov keywords in the keyword search task," in *2013 IEEE Workshop on Automatic Speech Recognition and Understanding*, 2013, pp. 416–421.