OLR-BIT Submission for OLR2021

Qingran Zhan[1]*, Chenguang Hu*[1]*,Xinmei Su*[1]*,Xiang Xie*[2]

[1] School of Information and Electronics, Beijing Institute of Technology, Beijing, China
[2] Shenzhen Research Institute, Beijing Institute of Technology, Shenzhen, China

qingran.zhan@gmail.com

Abstract

In this report, we describe our OLR-BIT submission for the OLR 2021 Challenge. We participate in constrained language identification (LID) track and automatic speech recognition tracks (ASR). Our submission system for LID is a fusion of multiple models. One model makes use of an attention-based fusion of PLP features and MFCC features, while the other models use MFCC features. The encoder networks we adopted are TDNN and ResNet. For the ASR task, an universal asr acoustic model is trained on multilingual dataset with shared hidden layer and then fine-tune the model for each specific languages. For the language models, RNN-based models are trained for specific languages. At decoding stage, the language label for test set is taken from LID, and using the language-depended model to decode. Experimental results indicate our method gets better performance than baseline, and character error rate is 12.79% on the development set.

Index Terms: speech recognition, language recognition

1. Introduction

The oriental languages include various language families, such as Austroasiatic languages, Tai-Kadai languages, Hmong-Mien languages, Sino-Tibetan languages, Altaic languages and Indo-European languages and it is an important part of the whole world's languages. As studied, the oriental languages themselves can effect each other and lead to complex evolution of languages.

The OLR 2021 Challenge is based on the last five challenges. It involves more languages and dialects and provide several new rules. This new challenge sets four tasks, two on LID and two on ASR: constrained LID (task1), unconstrained LID (task2), constrained multilingual ASR (task3) and unconstrained multilingual ASR (task4). We submitted the final results of the task1 and task3. The submission systems for LID contains different acoustic features, multiple data augmentation methods and different xvector models, the same back-end classifier is applied. Finally, average fusion strategy is used to fuse different models. For ASR task, we select the Conformer as the acoustic model and RNN as the language model as the language-depended training.

2. Method

Several systems are trained using the datasets defined for the final system design. Experiments are performed with the toolkit Pytorch and Kaldi.

2.1. Language identification

2.1.1. Training recipe

The final training recipe includes:

Table 1: *The architecture of ResNet34*

Layer	Structure	Stride	Output
Input	-	-	$64 \times 100 \times 1$
Conv2D-1	$3 \times 3, 32$	1	$64 \times 100 \times 32$
ResNetBlock-1	$\begin{bmatrix} 3 \times 3, 32 \\ 3 \times 3, 32 \end{bmatrix} \times 3$	1	$64 \times 100 \times 32$
ResNetBlock-2	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 4$	2	$32 \times 50 \times 64$
ResNetBlock-3	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 6$	2	$16 \times 25 \times 128$
ResNetBlock-4	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 3$	2	$8 \times 13 \times 256$
Statistics Pooling	-	-	16×256
Flatten	-	-	4096
Linear1	-	-	256
Linear2	-	-	13

- the additive margin loss (AM) function
- the four data augmentation strategies mentioned in Subsection 3.2.2
- attentive pooling
- attention-based fusion for MFCCs and PLP, which is shown in Figure 1
- using ResNet instead of TDNN
- using the SPP [1] method to paste the xvectors

2.1.2. Back-end classifiers

The back-ends consist of linear discriminant analysis (LDA) dimension reduction, whitening, centering, length normalization and logistic regression (LR).

2.1.3. Fusion

For score level, the fusion method is used by computing the average of the scores of the individual systems.

2.2. Automatic speech recognition

2.2.1. Training recipe

The final training recipe includes:

- the conformer-based model [2] with a 12 layers encoder and 6 layers decoder
- RNN language model
- data augmentation strategies mentioned in Subsection 3.2.2

3. Experiments

3.1. Corpora

For the dataset used in this challenge is all provided in the paper [3].

3.2. Language identification

In this section, we report data preprocessing and model configuration of our LID system.

3.2.1. Preprocessing

Before training, several methods of data augmentation are adopted, including speed perturbation, volume augmentation, spectral augmentation and noise adding. For speed perturbation, a speed factor of 0.9 or 1.1 to slow down or speed up the original recording is applied. Two augmented copies of the original recording is added to the original dataset to obtain a 3-fold combined training set. For volume augmentation, the volume is increased or decreased randomly. For noise adding, the Musan dataset [4] is used to add several noises.

Two kinds of frame-level features are implemented, one is MFCC features and the other is the PLP features. The 20-dimensional MFCC features and 3-dimensional pitch features are extracted and combined together. The features are computed by Kaldi [5] with 25 ms window length and 10 ms shift. Furthermore, the 20-dimensional PLP features are extracted.

3.2.2. Model Configuration

Two neutral networks are considered as the model to train xvector system: Time-delay Neural Network (TDNN) and ResNet34 [6]. For the TDNN, we follow the standard recipe in OLR2021 challenge baseline. ResNet34 is a successful DNN architecture developed for image processing and used in a wide variety of tasks, including speech processing. We adopt the ResNet34 in our experiments where the architecture is depicted in Table 1.

The MFCC-Pitch Features and the PLP features are pasted into 43-dimensional features in order to use the attention mechanism to fuse the features in the network training.

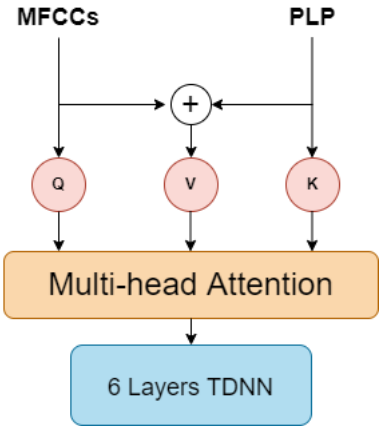

Figure 1: *The attention based fusion of MFCC and PLP.*

3.2.3. Results

Five models are trained using the recipe mentioned above in our experiments. The testing results are as in Table 2.

Table 2: *Results of the experiment systems - C_{avg}, which are evaluated on development set.*

Input Features	**Models**	C_{avg}
MFCC	TDNN(statistic pooling)	0.0268
MFCC	ResNet	0.0355
MFCC & PLP	TDNN(statistic pooling)	0.0298
MFCC	TDNN (attentive pooling)	0.0288
MFCC	TDNN(SPP& statistic pooling)	**0.0267**
-	Fusion	**0.0267**

In Table 2, a TDNN network and the SSP method achieve the lowest C_{avg}, other network also have good performance in the experiment. Furthermore, after we adopt the fusion method, the C_{avg} also achieve the lowest score, but it doesn't seem to decrease comparing with the TDNN network using the SPP method.

3.3. Automatic speech recognition

In this section, we report data preprocessing and model configuration of our ASR system.

3.3.1. Preprocessing

The acoustic features are composed of 80-dimensional Mel Filterbank and 3-dimensional pitch feature. We also adapt specaugment, noise adding and speed perturbation to the training set to for data enhancement. All the feature preprocessings are conducted using Kaldi and then we train the end-to-end ASR model and language model on the Espnet toolkit [7].

3.3.2. Model Configuration

In this report, we choose Conformer to model the end-to-end ASR system, and for the language model, RNN-based architecture is applied. For the acoustic model, an universal acoustic model on multilingual dataset is trained and we adapt the model with each language firstly. The language models are trained using the text from training set for each languages. The final model is taken from the average of the last 15 models. Since all the acoustic models and language models are language-depended so finally we get 13 AM-LM models. At decoding stage, we first pass the test data to LID system to get the language information and use the language-specific model to decode.

3.3.3. Results

The results are shown in Table 3, it can be found that our system has achieved a character error rate of 12.79% on the development set. Compared with the baseline system, the system has improved in all languages.

4. Conclusions

In this paper, we illustrate the details of our submitted systems for OLR 2021 challenge. For LID task1, the final submitted systems are a fusion of five systems and achieve a C_{avg} of 0.0267. For ASR tasks, the final submitted system is a composed of language-specific acoustic and language model. At decoding stage, the model uses the language information acquired by the LID system. On the development dataset, our submitted system achieves a CER of 12.79%.

Table 3: *CER comparison of different methods on OLR 2021 task3 dev set.*

Language	Total	zh-cn	Minnan	Shanghai	Sichuan	ct-cn	id-id	ja-jp	ko-kr	ru-ru	Kazak	Tibet	Uyghu
Baseline	35.5%	26.6%	64.8%	33.8%	22.7%	24.7%	21.8%	33.1%	34.4%	40.2%	15.5%	21.0%	31.8%
Ours	12.8%	16.1%	58.7%	26.6%	15.3%	14.0%	8.5%	17.5%	19.5%	22.5%	9.9%	7.4%	6.1%

5. References

[1] S. Y. Z. L. H. L. L. L. M. Zhao, R. Li and Q. Hong, "Phone-aware multi-task learning and length expanding for short-duration language recognition," *APSIPA*, 2019.

[2] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu, and R. Pang, "Conformer: Convolution-augmented transformer for speech recognition," 2020.

[3] J. L. Y. Z. Z. L. Q. H. L. L. D. W. L. S. C. Y. Binling Wang, Wenxuan Hu, "Olr 2021 challenge: Datasets, rules and baselines," 2021.

[4] D. Snyder, G. Chen, and D. Povey, "Musan: A music, speech, and noise corpus," 2015.

[5] 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 *IEEE 2011 workshop on automatic speech recognition and understanding*, no. CONF. IEEE Signal Processing Society, 2011.

[6] S. R. J. S. Kaiming He, Xiangyu Zhang, "Deep residual learning for image recognition," *CVPR*, 2015.

[7] S. Watanabe, T. Hori, S. Karita, T. Hayashi, J. Nishitoba, Y. Unno, N. E. Y. Soplin, J. Heymann, M. Wiesner, N. Chen, A. Renduchintala, and T. Ochiai, "Espnet: End-to-end speech processing toolkit," 2018.