Funspeech submission for OLR 2021 Challenge

Xudong Wang, Dachuan Zheng, Ming Xu

YunshangQulv, Beijing, China

{xudong.wang,dachuan.zheng,ming.xu}@ilivedata.com

Abstract

This paper describes the submitted system by funspeech for OLR 2021 challenge. The challenge contains four tasks: (1) constrained Language Identification (LID), (2) unconstrained LID, (3) constrained multilingual Automatic Speech Recognition (ASR), (4) unconstrained multilingual ASR. We participated in task (1), (2) and (3). The main points where our systems differ from the baseline are described in detail:

1. For task1 and task2, more data augmentation and acoustic bottleneck feature were applied in our experiments. And we directly used a softmax layer rather than x-vector to get the language identification results. The Top-N model fusion strategy was also implemented.

2. For task3, we used an end-to-end conformer model to train our ASR system.

Index Terms: language identification, automatic speech recognition, end-to-end

1. Introduction

LID refers to identifying the language categories from the given utterance, ASR refers to converting human speech into text. The OLR 2021 challenge[1] includes four tasks, two on LID and two on ASR: (1) constrained LID, (2) unconstrained LID, (3) constrained multilingual ASR, (4) unconstrained multilingual ASR. We participated in task1、task2 and task3, and submitted the results of these tasks according to the required test conditions by the challenge.

In this paper, we introduce the system of funspeech team for OLR 2021 challenge in detail. The remainder of this paper is organized as follows. Section 2 describes the data preparation. Section 3 introduces the approaches adopted in our systems. The experimental settings and results on the progress set are shown in Section 4. Finally, the conclusion is given in Section 5.

2. Data Preparation

In this OLR 2021 challenge, only the data provided by the organizer can be used for task1 and task3. The permitted resources are several specified data sets, including OLR16-OL7, OLR17-OL3, OLR17-dev, OLR17-test, OLR18-test, OLR19-dev, OLR19-test, OLR20-dialect and OLR20-test. And we used all the data in all the tasks. For task2, 17 languages are involved, and English is not included in provided data, we added about 10 hours English data shuffle chosen from GigaSpeech[2].

2.1. Augmentation

Except speed and volume perturbations, adding gaussian noise and SpecAugment[3] were also used to augment the training data for task1 and task2. We set signal to noise ratio (SNR) random from 10 to 30 to add gaussian noise. In SpecAugment, we used two ways to distort the signal. First, the signal was transformed into the spectrogram, followed with simultaneous usage of both masking approach: (i) time masking, where several randomly selected frames were replaced by spectrogram mean value; (ii) frequency masking, where several randomly selected frequency bins were replaced by spectral mean value across all frames. After the spectral masking procedure, the spectrogram was transformed back to signal, followed by feature extraction. Randomly selecting roughly 3 times the size of the original data from the augmentation data got better results than using the whole augmentation data in our system.

In task3, we only used speed perturbations and the SpecAugment.

2.2. Features

We used 80-dimensional filter banks with the 3-dimensional pitch as acoustic features to train ASR end-to-end model, and the ASR encoder output as the features for the LID tasks.

3. System Description

In this section, we will briefly describe our approaches for this challenge, including end-to-end ASR, LID system and results fusion.

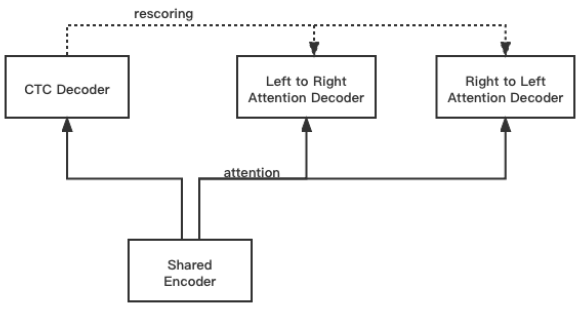

Figure 1: *WeNet U2++ architecture*

3.1. End-to-end ASR

In task3, we first got the train data and dict using baseline system[1][4], then we built our end-to-end model by WeNet[5] toolkit. Model architecture, as shown in Figure 1, contains four parts: a Shared Encoder that models the context of the acoustic features, a CTC Decoder that models the alignment of the frames and tokens, a Left to Right Attention Decoder that models the left tokens dependency, and a Right to Left Attention Decoder that models the right tokens dependency.

The Shared Encoder consists of multiple Conformer encoder layers. The CTC Decoder consists of a linear layer and a log softmax layer. The CTC loss function is applied over the softmax output in training[6].

We used the standard configuration of WeNet conformer which contained 12-layer encoder and 6-layer decoder with 2048-dimensional each layer. The attention sub-layer was 256-dimensional and used 4 attention heads. The CTC loss and Attention loss were combined in training.

3.2. LID system

In task1 and task2, we used TDNN baseline system[1][7]. We got better result when we change data chunk from 1s to 3s during training and discrimination. Using softmax also brought a slight improvement. In the later experiments, we used a softmax layer rather than x-vector to get the score. The flow chart of LID is shown in Figure 2.

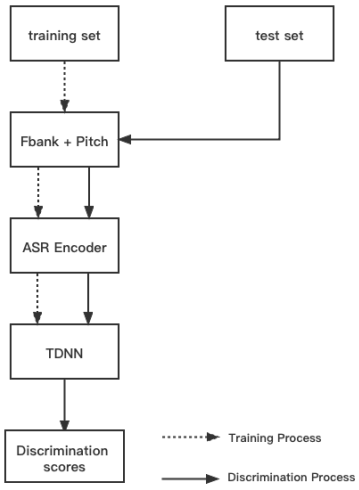

Figure 2: *The flow chart of LID*

3.3. Results fusion

The fusion strategy which calculates weighted average of the top n-best results to obtain the final result was adopted in our systems. We chose 4 best results for fusion in task1 and task2. The fusion weight were set to the same fusion, and the sum of fusion weight was 1.

4. Experimental Settings and Results

4.1. Experimental Settings

For task3, we trained our end-to-end ASR system on 2 NVIDIA GeForce GPUs with Adam optimizer and warm-up was used for the first 25000 iterations. The whole network was trained for 40 epochs. Best 10 models were fusion to obtain the final model. We used CTC greedy search to get the final hypotheses.

For task1 and task2, we first got the train features from encoder output by ASR model, which got a better result than MFCC or Fbank with Pitch features. For task1, there was only 13 languages to identify with 17 languages in train data, so we added masks for the other 4 languages scores. For task2, English is not included in provided data, so we added English data and used masks to the scores too.

The results are presented in the Table 1 and Table 2.

Table 1: *The constrained LID results on Progress sets*

	Cavg*	EER[%]
Baseline	0.0823	9.0380
chunk_300	0.0584	6.4210
data_augment	0.0370	4.0290
Bottleneck features	0.0106	1.1560
Results fusion	0.0080	0.9129

chunk_300: Change data chunk from 1s to 3s during training and discrimination.

data_augment: Data augmentation.

Bottleneck features: Use ASR Encoder output as the TDNN input features.

Results fusion: Best results fusion. Final submission result.

Table 2: *The constrained multilingual ASR results on Progress sets(CER%)*

	TOTAL *
Baseline	39.1
WeNet	31.4

4.2. Experimental Results

Table 1 shows the main results of our constrained LID systems on the referenced progress sets in comparison with the baseline systems.

As shown in Table 1, using 3s training and discrimination data got better result than 1s. Data augmentation was also benefit for our system. Using ASR Encoder output as the TDNN input features got the best result in single system. Finally, results fusion could further improve performance.

Table 2 shows the results of our end-to-end ASR with the baseline systems.

5. Conclusion

In this paper, we presented the funspeech system for the OLR 2021 challenge. We described the data processing, used architectures, experimental settings and results.

6. References

1. B. Wang, W. Hu, J. Li, Y. Zhi, Z. Li, Q. Hong, L. Li, D. Wang, L. Song, and C. Yang, “OLR 2021 Challenge: Datasets, Rules and Baselines”, 2021.
2. G. Chen, S. Chai, G. Wang, et al. “GigaSpeech: An Evolving, Multi-domain ASR Corpus with 10,000 Hours of Transcribed Audio”, arXiv preprint arXiv:2106.06909, 2021.

3. D. S. Park, W. Chan, Y. Zhang, C. C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le. “Specaugment: A simple data augmentation method for automatic speech recognition”, arXiv preprint arXiv:1904.08779, 2019.
4. S. Watanabe, T. Hori, S. Karita, T. Hayashi, J. Nishitoba, Y. Unno, N. Soplin, J. Heymann, M. Wiesner, and N. Chen, “Espnet: End-to-end speech processing toolkit”, 2018.
5. Z. Yao, D. Wu, X. Wang, et al. “Wenet: Production oriented streaming and non-streaming end-to-end speech recognition toolkit”, 2021.
6. D. Wu, B. Zhang, C.Yang, Z. Peng, W, Xia, X. Chen, X. Lei, “U2++: Unified Two-pass Bidirectional End-to-end Model for Speech Recognition ”, 2021
7. Z. Li, M. Zhao, J. Li, Y. Zhi, L. Li, Q. Hong. “The XMUSPEECH System for the AP19-OLR Challenge”, 2020.