Discription for AP20-OLR

Li Lihan[1], *Yang Shuo*[2]

[1]Anonymous
[2]Anonymous
xjpgjzl@gmail.com

Abstract

In this paper, we present our system for the oriental language recognition (OLR) challenge, AP20-OLR. The challenge this year contained three tasks: (1) cross-channel LID, (2) dialect identification, and (3) noisy LID. We leveraged the system pipeline from two aspects, including front-end training and fusion strategy. We implemented two encoder networks for Task1 and 3.

Index Terms: AP20-OLR, language identification, fusion

1. Introduction

The AP20-OLR challenge included three tasks. Task 1 involved cross-channel LID which means the language of each utterance is among the known traditional 6 target languages, but utterances were recorded with different channels. Task 2 was dialect identification in which three nontarget languages are added to the test set with the three target dialects. Task 3 was noisy LID where noisy test data of the 5 target languages will be provided. All tasks were evaluated and ranked separately. The principle evaluation metric was Cavg, which was defined as the average of the pairwise performance of test languages, given Ptarget = 0.5 as the prior probability of the target language.

We submitted the final results of task 1 and task 3. Our developed systems consisted of front-end training and fusion strategy.

2. Data Preparation

In this AP20-OLR challenge, additional training materials were forbidden to participants, and the permitted resources were several specified data sets, including AP16-OL7, AP17-OL3, AP17-OLR-test, AP18-OLR-test, AP19-OLR-dev, AP19-OLR-test, AP20-OLR-dialect and THCHS 30.

2.1. Language Identification Training Set

Before training, we adopted the data augmentation, including speed perturbation, to increase the amount and diversity of the training data. For speed perturbation, we applied a speed factor of 0.8 or 0.9 or 1.1 or 1.2 to slow down or speed up the original recoding. Four augmented copies of the original recoding were added to the original data set to obtain a 5-fold combined training set.

For task1, AP16-OL7, AP17-OL3, AP17-OLR-test, AP18-OLR-test and THCHS 30 constituted the training set. The training set used in Task1 was named AP20-task-1-train, which only included the six target languages. Based on previous competition experience, we downsample the audio data with a sampling rate of 16000 to 8000, and then upsample it to 16000 to adapt the cross-channel task.

For task3, in AP16-OL7, as for Russian, Korean and Japanese, there are 2 recoding sessions for each speaker: the first session was recoded in a quiet environment and the second was recoded in a noisy environment. VAD was used in the data which was recoded in a noisy environment. We named the noisy part from the result of VAD the noisy-training-set. It was randomly selected to mix into the training set which can better match the noisy LID.

2.2. Enrollment Set

The enrollment set for Task1 was AP19-OLR-test in which only the six target languages were remained. For task3, the noisy-training-set mentioned in 2.1 was randomly selected to mix into the AP19-OLR-test in which only the five target languages were remained.

3. System Discription

3.1. Feature Extraction

The 40-dimensional Filter Banks(FBanks) was used as the acoustic features. All features had frame-lengths of 25ms, frame-shifts of 10ms, and mean normalization over a sliding window of up to 3 seconds. The energy VAD was used to filter out non-speech frames. The feature engineering was executed using the Kaldi platform.

3.2. Encoder Networks

Two different encoder networks were conducted for Task1 and 3 based on the Pytorch platform. We also tried different encoder networks, but no significant improvement was achieved in this challenge.

3.2.1. Res2netNonLocalGruAttention

The proposed Res2netNonLocalGruAttention model is shown in Fig.1. The training chunk size between 100-150 was used in the sequential sampling when prepared the training examples. The model was optimized with Adam optimizer, with a mini-batch size of 512.

3.2.2. Res2netGCnetGruVLAD

The proposed Res2netGCnetGruVLAD model is shown in Fig.2. Compared with the Res2netNonLocalGruAttention model, this model replaces the Non-local block with GCNet and the attention pooling with NetVLAD. The loss function was replaced by AM-softmax and everything else was the same.

3.3. Score Fusion

We use the above two encoder networks to classify and get two different scores which were fused by different weight.

The enrollment set was used to get α, β. When the Cavg on the enrollment set is the lowest, α, β at this time were used as

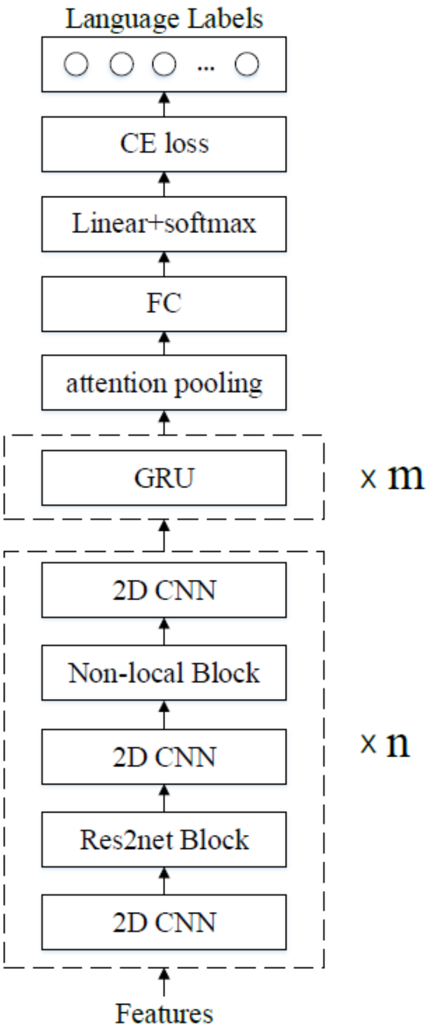

Figure 1: *Res2netNonLocalGruAttention*

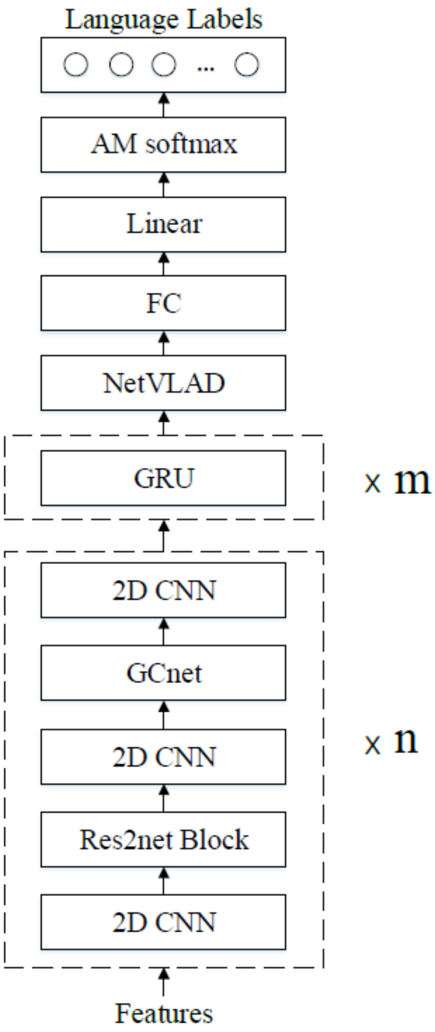

Figure 2: *Res2netGCnetGruVLAD*.

the final weight.

$$Score = \alpha Score_1 + \beta Score_2 \qquad (1)$$