The NISP System for the OLR 2021 Challenge

Junjie Jin, Wenxuan Wang, Binbin Du, Yuting Yang, Yuke Li

YiDun AI Lab, Netease, Hangzhou 310052

{jinjunjie, wangwenxuan, dubinbin,yangyuting04, liyuke}@corp.netease.com

Abstract

In this report, we describe the submissions of Yidun NISP for all of the four tasks. For task1, we fuse ResNet with Squeeze-Excitation and Ecapa-tdnn to achieve 0.0417 Cavg. For task2, we fine-tune the wav2vec2.0 encoder with additional pooling and linear layer. We also expand the training dataset by collecting data from common voice. For task3, we use a hybrid CTC/Attention end-to-end approach to model Multilingual-ASR tasks, and meanwhile try to select and fuse acoustic and language models in multiple ways based on language classification information. For task4, we collect additional open source data and try cross-language transfer learning from high-resource languages to low-resource languages. Finally, the performance of our system exceeds the baseline greatly.

Index Terms: language identification, deep neural network, Multilingual-ASR

1. Introduction

OLR Challenge is the sixth oriental language recognition (OLR) challenge with four tasks: (1) Constrained Task cross-channel language identification, which includes 13 target languages (Indonesian, Japanese, Russian, Korean, Vietnamese, Mandarin, Cantonese, Sichuanese, Shanghainese, Hokkien, Tibetan, Kazakh and Uyghur), (2) Unconstrained Task language identification, which involves 17 languages (Indonesian, Japanese, Russian, Korean, Vietnamese, Thai, Malay, Telugu, Hindi, English, Kazakh, Tibetan, Uyghur, Mandarin, Sichuan, Shanghainese, Hokkien) and can use any data and pertained model, (3) Constrained ASR, which only the data provided by the organizer can be used and (4)Unconstrained ASR, where any publicly labeled can be used for training and optimization. Task1 and task2 are Language Identification (LID) tasks. Task3 and task4 are multilingual ASR tasks. The following sections describe submitted systems for all the tasks.

2. Data definition

For task1, OLR16, OLR17, OLR18, OLR19-dev, OLR20-dailect, namely task1-train constitute the training set.OLR20-test and OLR2019-test which were named $task1 - enroll$ for enrollment sets.

For task2, we used,Telugu, Hindi Chinese, English from Common voice[1]. The train and enroll dataset for task 2 is named, $task-2-train-cv-aug$ and $task-2-enroll-cv-aug$ respectively.

For task3, we only use official data, and split the training set and validation set according to the baseline[2].

For task4, in addition to official data, we additionally use the ASR open source data sets from openslr and magichub for some languages. The external data sets used are shown in the following table 1:

Table 1: *External dataset list of task4*

language	Dataset name	Source	Duration/h
Mandarin	Aishell	SLR-33	151
Mandarin	Free ST	SLR-38	110
Mandarin	Primewords	SLR-47	99
Mandarin	aidatatang_200zh	SLR-62	140
Mandarin	MAGICDATA	SLR-68	712
Japanese	JSSC	magichub	18
Korean	Zeroth-Korean	SLR-40	51.6
Kazakh	Kazakh Speech Corpus	SLR-102	332
Russian	Golos	SLR-114	1240

3. Experiment setup

3.1. Augmentation

Several data augmentation methods is used to the raw data pipeline (reverb, music, noise, speed perturbation(sp) (0.9x 1.1x)[3]), also spectral augmentation (SpecAugment)[4] with hyper parameters of freq-max-proportion=0.3, time-zeroed-proportion=0.2, and time-mask-max-frames=20 is applied.

3.2. Features

For task1 and task2, we used 64-dimensional filter banks. The filter banks were computed in Kaldi with 25 ms window length and 10 ms shift. For task2, raw wave is feed to the model. For both task1 and task2 utterance-level mean and variance normalization (CMVN) is applied for robustness.

For task3 and task4, the input of the acoustic model is 80-dimensional LogFbank concatenated with 3-dimensional pitch features. We use character as modeling unit of Mandarin, Shanghainese, Sichuanese, Hokkien and Cantonese. And subwords encoded by BPE[5] are employed as the multilingual modeling unit of the other languages.

3.3. LID Architecture

3.3.1. Pipeline

For task1 and task2, in order to keep the context information as much as possible, we dynamically batch the training set and group utterance based on its duration in which of 0-1, 1-2,. . . ,29-30 seconds chunks and randomly subsample the utterance to 30s if longer than 30s.

3.3.2. task1

Resnet34SE[6], a variant of Resnet which adds Squeeze-and-Excitation block (SE-block) to Resnet, which is a successful DNN architecture developed for image processing and used in a wide variety of tasks, including speech processing.

Ecapa-tdnn[7] is a neural net architecture popularly used in

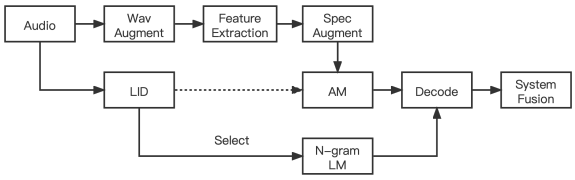

Figure 1: *ASR system framework*

speech processing for tasks like SID or LID. The deep structure was trained to classify the N languages using the cross entropy (CE) loss function. The final system is constituted of the fusion of two models: resnetSE and Ecapa-tdnn with equal wights, named $Fusion$.

3.3.3. task2

Pretrained wav2vec2.0(w2v-encoder)[8] is believed to have the ability to capture the information about the speaker and language.[9] w2v-encoder consists of CNN-based feature encoder, a Transformer-based context network. We add a statistics pooling layer and a fully connected layer to w2v-encoder. The CNN encoder of w2v-encoder is freezed during finetune.

3.4. ASR Architecture

3.4.1. Workflow

Our system framework is shown in Figure 1.Since the language information of each audio in the test set is unknown, we train a multilingual speech recognition model with a single end-to-end model for all languages. At the same time, we selected a specific language model to participate in the decoding and re-scoring based on the LID information such as the results of task1 and task2. On the other hand, we train a separate acoustic model for several languages with serious confusion in the recognition results and combine language information for system integration.

3.4.2. Language identification

Before speech recognition, we use the methods and results of task1 and task2 respectively as the language classification information of task3 and task4.

3.4.3. Acoustic model

The acoustic model adapts the conformer structure[10], which has 12-layer encoder with 2048 units and a 6-layer decoder with 2048 units. There are 4-head attention with 256 dimensions. The in lambda in Equation 1 is 0.5. During decoding, the beam size is 20. We have used the data of all languages to train a unified acoustic model, as well as individually trained acoustic models for specific languages. We perform model selection and model fusion based on the comparison of language classification information and the results of dev and progress sets which is better. The ASR loss is based on a hybrid CTC/Attention structure[11]. The output of the encoder is used to calculate the CTC loss, and the output of the decoder together with the ground-truth label are utilized to obtain the CE loss. During training, the loss functions of the two branches will be linearly combined in a certain proportion. As shown in Equation 1 , where lambda denotes the weight of different loss.

Table 2: *Results on task1 and task2 systems*

Task	Model	EER%	Cavg
task1	Resnet34SE	6.86	0.0617
task1	Ecapa-tdnn	8.39	0.0793
task1	Fusion	5.02	0.0417
task2	wav2vec+pooling	15.31	0.1612

Table 3: *CER of task3 and task4 systems*

Language	Code	Baseline	Task3	Task4
Hokkien	Minnan	64.1	60.7	53.4
Shanghainese	Shanghai	34.0	28.9	22.1
Sichuanese	Sichuan	22.8	15.5	11.1
Cantonese	ct-cn	19.3	16.0	12.0
Mandarin	zh-cn	27.0	25.7	16.0
Japanese	ja-jp	30.0	28.7	18.0
Korean	ko-kr	29.1	24.3	8.9
Indonesian	id-id	16.4	13.4	13.4
Russian	ru-ru	32.2	28.4	23.1
Vietnamese	vi-vn	10.9	7.9	7.1
Kazakh	Kazak	15.8	11.1	8.9
Tibetan	Tibet	27.3	9.8	9.4
Uyghur	Uyghu	13.0	8.1	8.0
Avg-CER(%)		21.6	16.0	13.3

$$loss = \lambda loss_{ctc} + (1-\lambda) loss_{att} \quad (1)$$

During inference, the Encoder-CTC branch generates candidate sequences through beam search decoding, and the Decoder-Attention branch re-scores and sorts the candidate sequences to obtain the optimal sequence.

3.4.4. Language model

In this paper, the language model doesn't use the text of all languages to generate a common language model, but uses each language text to generate a specific 4-gram language model. And the language model is selected according to the results of LID when decoding. We find that compared with a single multilingual speech model, this method can effectively alleviate the confusion between languages in the ASR recognition results

3.4.5. Cross-lingual transfer learning

In the task4, we added external open source data for training. There are big differences in the amount of open source data available in different languages. For example, Mandarin has lots of open source corpus, while dialects have almost none. Therefore, we try cross-language transfer learning for dialect based Mandarin ASR model, which can significantly improve the performance of several dialects.

4. Results

As shown in Table 2, our system achieves EER of 5.02% and Cavg of 0.0417 on $task1-enroll$ set of Task1, EER of 15.31% and Cavg of 0.1612 on $task-2-enroll-cv-aug$ of Task2.

Because the test channel of the progress set has been closed when the final result is completed, the result of task3 and task4 under dev is shown in Table 3.

5. Conclusion

In this work, we presented our language identification and ASR system for OLR 2021 Challenge. And our fusion submission achieved 0.0417 Cavg and 0.1612 Cavg for task 1, and task 2 on our own development set respectively.

We use a hybrid CTC/Attention end-to-end approach to model multilingual ASR tasks, and meanwhile try to select and fuse acoustic and language models in multiple ways based on language classification information. It can effectively alleviate the confusion between different languages and greatly improve the recognition performance of ASR. In an unrestricted scenario, facing the imbalance of data resources between different languages, we try cross-language transfer learning from high-resource languages to low-resource languages, which can effectively improve the recognition performance of low-resource languages. In the future, we will further develop other multilingual modeling methods under low-resource conditions (such as UPS) and a more effective combination of ASR and LID.

6. References

[1] R. Ardila, M. Branson, K. Davis, M. Henretty, M. Kohler, J. Meyer, R. Morais, L. Saunders, F. M. Tyers, and G. Weber, "Common voice: A massively-multilingual speech corpus," *arXiv preprint arXiv:1912.06670*, 2019.

[2] 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," *arXiv preprint arXiv:2107.11113*, 2021.

[3] T. Ko, V. Peddinti, D. Povey, and S. Khudanpur, "Audio augmentation for speech recognition," in *Sixteenth annual conference of the international speech communication association*, 2015.

[4] 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.

[5] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," *Computer Science*, 2015.

[6] J. S. Chung, J. Huh, S. Mun, M. Lee, H. S. Heo, S. Choe, C. Ham, S. Jung, B.-J. Lee, and I. Han, "In defence of metric learning for speaker recognition," in *Interspeech*, 2020.

[7] B. Desplanques, J. Thienpondt, and K. Demuynck, "Ecapa-tdnn: Emphasized channel attention, propagation and aggregation in tdnn based speaker verification," *arXiv preprint arXiv:2005.07143*, 2020.

[8] A. Baevski, H. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," *arXiv preprint arXiv:2006.11477*, 2020.

[9] Z. Fan, M. Li, S. Zhou, and B. Xu, "Exploring wav2vec 2.0 on speaker verification and language identification," *arXiv preprint arXiv:2012.06185*, 2020.

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

[11] S. Watanabe, T. Hori, S. Karita, T. Hayashi, J. Nishitoba, Y. Unno, N. E. Y. Soplin, J. Heymann, M. Wiesner, N. Chen *et al.*, "Espnet: End-to-end speech processing toolkit," *arXiv preprint arXiv:1804.00015*, 2018.