Huawei Amsterdam OLR 2021 Submission

Dimitrios Bountouridis, Jun Luo, Dmytro Balaban, Robin Cahierre, Yang Sun

Huawei Research Center, Amsterdam, The Netherlands

{dimitrios.bountouridis,junluo,balaban.dmytro,robin.cahierre,sunyang48}@huawei.com

Abstract

In this paper, we present our system for the oriental language recognition challenge, OLR2021. The challenge this year contained four tasks: (1) constrained language identification (LID), (2) unconstrained LID, (3) constrained multilingual automatic speech recognition (ASR), (4) unconstrained multilingual ASR. Our submission addresses tasks (1), (3) and (4). For task 1 we adopted the transformer model architecture by incorporating a language token in a sequence-to-sequence learning paradigm. We trained two systems using different features and finally combined their outputs. For task 3, we relied on a single end-to-end multilingual WeNet model trained on noise-augmented data. Finally for task 4, we developed a cascaded approach by first performing language identification and then combining the predictions of various wav2vec, WeNet monolingual and multilingual models either readily available online or trained by our team. Our systems showed varying performance on the evaluation sets, largely depending on the language.

Index Terms: speech recognition, language identification

1. Introduction

Language Identification (LID) refers to identifying the language of human speech, and it is usually presented at the front-end of other speech processing systems, such as the equally important Automatic Speech Recognition (ASR) i.e., the conversion of human speech into text. While LID and ASR for Western languages have been largely studied, oriental languages have been under-explored. Most interestingly, with the worldwide population movement and communication, various multilingual phenomena have emerged resulting in oriental languages influencing each other via multilingual interaction.

To encourage improving the research on multilingual phenomena and advancing the development of multilingual speech technologies, the oriental language recognition challenge has been organized annually since 2016. For OLR2021, four tasks were proposed. Task 1 constrained LID, is described as "a cross domain identification task with constrained training condition" and it pertains to identifying the language of around 32000 utterances from 13 languages, while using only the training data provided by the OLR organizers. Task 2 unconstrained LID, is described as "a cross domain identification task with unconstrained training condition" with the test data comprising real-life environments. In practice, Task 2 extends to 17 languages and allows for using external data or models to solve the problem. Task 3 constrained multilingual ASR pertains to speech recognition on the same 32k utterances from Task 1, while allowing using only the data provided by the organizer for the purposes of training acoustic and language models. Finally, for the Task 4 of unconstrained multilingual ASR, any data is allowed to be used to train the acoustic and language models.

For Tasks 1 and 2 the principle evaluation metric was C_{avg}, which was defined as the average of the pairwise performance of test languages, given $P_{target} = 0.5$ as the prior probability of the target language. For Tasks 3 and 4 the evaluation metric was Character Error Rate (CER): the sum of deletion, insertion, and substitution errors in the ASR output compared to the reference transcription, divided by the total number of characters in the reference transcription.

Our team "Huawei Amsterdam" attended the OLR challenge for the first time this year and applied for three of the four tasks (1), (3) and (4). The rest of this paper is organized as follows: Section 2 describes the data preparation process for each task. Sections 3, 4 and 5 describe the methods used to build the systems for Tasks 1, 3 and 4 respectively. The results for the development set are shown in Section 6. Finally, we present some concluding remarks in Section 7.

2. Data preparation

2.1. Constrained LID and ASR Training Set

For Tasks 1 and 3, additional training materials were forbidden to participants, and the permitted resources were several data sets from previous OLR challenges extended with new languages. The data sets included thirteen languages, which were Mandarin (zh-cn), Cantonese (ct-cn), Indonesian (id-id), Japanese (ja-jp), Russian (ru-ru), Korean (ko-kr), Vietnamese (vi-vn), Kazakh (Kazak), Tibetan (Tibet), Uyghur (Uyghu), Shanghainese (Shanghai), Sichuanese (Sichuan) and Hokkien (Minnan). The data was arranged into train and development sets of sizes around 105k and 4k utterances respectively. For the sake of simplicity, we call this data the OLR set.

Before training, we adopted noise addition as a type of data augmentation to increase the amount, diversity and robustness of the data. We added a noise sample from the MUSAN collection [1] to each recording at an SNR level randomly selected from $[0, 5, 10, 15]$ dB. So finally, we had one augmented copy of the data which was added to the original to obtain a 2-fold combined train and development set.

2.2. Unconstrained ASR Training Set

For Task 4, we were not constrained to the OLR set. We therefore decided to additionally use any freely available online sources from Commonvoice[1] and/or OpenSLR[2]. We limited ourselves to those data repositories due to time constraints. In Table 1, we present the amount of data in hours that we found for some of the OLR languages. No noise addition was applied to this collection of data.

2.3. Transcriptions preparation

Both OLR and freely available data from Commonvoice or OpenSLR came with noisy transcriptions i.e., they were inconsistently cased, included punctuation marks and unneces-

[1] commonvoice.mozilla.org/en/datasets

[2] openslr.org

Language	CommonVoice	OpenSLR
id-id	23h	-
ja-jp	26h	-
Kazak	0.7h	-
ru-ru	148h	-
vi-vn	3h	-
ko-kr	-	51.6h

Table 1: *Amount of hours of freely available data for six of the thirteen OLR languages.*

sary tags. Prior to training we performed some text cleaning to alleviate some of those issues. However, the processing was language-dependant since each language has its own particularities. Unfortunately, for some languages such as Tibetan, preprocessing was based on merely intuition since our team did not include any native Tibetan speakers.

3. Constrained LID: System Description

For the language identification of Task 1 we employed transformer models. The setup was a typical sequence-to-sequence transformer configuration, where the inputs are extracted speech features, and the outputs are expected text transcripts; the only minor change was that language token was added to the beginning of the text transcripts, so the model also learns to predict the language. We use label smoothed cross entropy to optimize the model.

3.1. Feature Extraction and Text Tokenization

Features were extracted from 16kHz audio data. Two models were built with different features: the non-pitch system used 80-dimensional fbank, while the pitch system adds one additional pitch feature to the 80-dimention fbank feature. SpecAugment[2] was applied during training.

A character based tokenizer was trained using all the text transcripts from OLR training data set. In addition, thirteen language tokens were added to the dictionary as special tokens.

3.2. Likelihood Combination

The final likelihood was the combination of output from the pitch and non-pitch systems, weighted by inverse entropy. Formally, for the i-th system which outputs $P_i^{(j)}, j = 1, 2, ..., 13$ as the likelihoods for all 13 languages, its corresponding weight w_i was calculated as follows:

$$w_i = \frac{1}{\sum_{j=1}^{13} P_i^{(j)} \cdot [-\log P_i^{(j)}]} \tag{1}$$

The likelihood for an utterance being the j-th language was the weighted average of the two systems' likelihood:

$$P^{(j)} = \frac{\sum_{i=1}^{2} w_i \cdot P_i^{(j)}}{\sum_{i=1}^{2} w_i} \tag{2}$$

4. Constrained Multilingual ASR: System Description

For Task 3 our aim was to build a single end-to-end model that can perform multilingual ASR without the need of any LID prior knowledge.

4.1. WeNet

Our system is based on the WeNet end-to-end model; a U2 model [3] aiming to unify both streaming and non-streaming ASR. The setup of the WeNet model is similar to that of ESPnet [4] but with the few modifications most notably in the decoding strategy. As such WeNet can be described as a hybrid connectionist temporal classification (CTC)/attention architecture with transformer or conformer as encoder and an attention decoder to rescore the CTC hypotheses.

Two kinds of acoustic features were used: 80-dimensional fbank, with 3-dimensional pitch features. In addition to the augmentations applied prior to training (see Section 2.1), SpecAugment was applied during training and a global CMVN technique was applied on top of the features. Regarding transcription tokenization, WeNet uses SentencePiece[3] as its default tokenizer i.e., text splitting into sub-word units. Meanwhile, the OLR languages can be either word- or character-based. For example, modern Korean is written with spaces between the different words while Japanese is ultimately based on characters without spaces. To accommodate for all languages we used a char-based tokenization across all transcriptions which resulted to 6861 outputs for our multilingual model. The max trainable epoch was set to 120 and the last 20 epochs were averaged to form the final model.

5. Unconstrained Multilingual ASR: System Description

For Task 4 given its unconstrained nature and the small size of the original OLR training data, we decided to leverage whatever materials were freely available at the moment for the OLR languages. This includes both training data and pre-trained/ready-to-use models. Our hypothesis was that given a large amount of models with different predictions and a combination/fusion strategy, we should be able to achieve a better performance than any of the individual models.

We employed a cascaded approach (see Figure 1): we first used our LID system from Task 1. We then computed the ASR predictions using multiple monolingual or multilingual models including the WeNet model from Task 3 and various WeNet and wav2vec models either trained by us or freely available on websites such as HuggingFace[4]. The monolingual models would only be used when the LID output agreed with the model's language. Language models (LM) were built and integrated into the models for some of the languages depending on the text data availability. All the different models/configurations are presented in the Table 5. The final prediction was based on fusing/combining the individual predictions per utterance using multiple sequence alignment.

5.1. WeNet Models

For Task 4 we used two WeNet models. First, the multilingual model trained specifically for Task 3. This model outputs a prediction for every input recording. Secondly, a model for Mandarin (zh-cn) trained on the AISHELL-2[5] dataset that can be found on the public WeNet github repository[6]. The latter model is based on a unified conformer architecture with a con-

[3] github.com/google/sentencepiece

[4] huggingface.co/models?filter=wav2vec2

[5] aishell-eval.oss-cn-beijing.aliyuncs.com

[6] github.com/wenet-e2e/wenet/tree/main/examples/aishell2/s0#unified-conformer-result

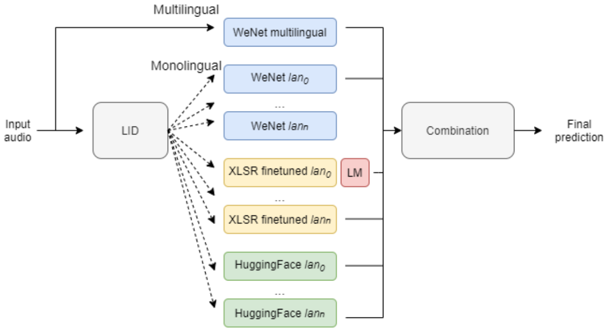

Figure 1: *Our cascaded approach for Task 4. The prediction of various monolinguals and multilingual ASR models is combined to form the final prediction. Monolingual models are triggered only when the LID prediction agrees with their corresponding language. For certain languages we use language models typically trained on Wikipedia data.*

former encoder and transformer decoder. It was trained with fbank features (no pitch) and no speed perturbation. We call this model "WeNet 20210421 unified conformer". In our pipeline, the model would produce a prediction only when the LID output was zh-cn.

5.2. Wav2Vec Models

Besides WeNet, we used the popular wav2vec 2.0 model [5]. We employed both pre-trained wav2vec models, for the purpose of finetuning, and already fine-tuned models.

Pre-trained models are implemented in fairseq [6] and can be found on the public fairseq github repository[7]. We made use of the multilingual pre-trained model XLSR-53, trained on 56k hours of unlabeled audio corresponding to 53 languages. XLSR-53 follows an architecture comprising 24 transformer blocks with model dimension 1,024, inner dimension 4,096 and 16 attention heads, resulting in a total of 300M parameters. XLSR-53 can be fine-tuned on labeled data using Connectionist Temporal Classification (CTC) [7] and a character-based output vocabulary. We finetuned XLSR-53 for all thirteen languages individually using two dataset configurations: 1) only OLR data and 2) OLR data plus Commonvoice or OpenSLR when available depending on the language.

Already finetuned wav2vec models are freely available on the HuggingFace model repository. We used seven finetuned models as shown in Table 2. It should be noted that for the Korean model we had to convert the predicted text from "jamo" characters to Hanguls to agree with the OLR format (Korean is an agglutinative language).

5.3. Language Models

We used Wikipedia[8] and the OpenSubtitles corpus[9] as our main sources of text for Indonesian (id-id), Vietnamese (vi-vn), Japanese (ja-jp), Mandarin (zh-ch), Korean (ko-kr) and Russian (ru-ru). For the rest of the languages we failed to locate either valuable text sources or compatible to the OLR format. For the available languages we trained separate KenLM[10] 5-gram ARPA language models that were used to decode the output of the corresponding wav2vec models.

Language	Name
id-id	indonesian-nlp/wav2vec2-large-xlsr-indonesian
vi-vn	nguyenvulebinh/wav2vec2-base-vietnamese-250h
ja-jp	jonatasgrosman/wav2vec2-large-xlsr-53-japanese
zh-cn	jonatasgrosman/wav2vec2-large-xlsr-53-chinese-zh-cn
ko-kr	fleek/wav2vec-large-xlsr-korean
ct-cn	ctl/wav2vec2-large-xlsr-cantonese
ru-ru	jonatasgrosman/wav2vec2-large-xlsr-53-russian

Table 2: *The collection of finetuned wav2vec HuggingFace models used in our pipeline. All are freely available at huggingface.co.*

5.4. Multiple Sequence Alignment

After all predictions have been computed for a recording, we combined/fused them to compute the "consensus" prediction using multiple sequence alignment (MSA). We used the MAFFT [8] toolkit but made sure that not all initial predictions were assigned equal weights when computing the MSA. Instead, we assigned weights to each prediction relative to their overall similarity to the rest.

6. Results

The performance of our systems on the OLR test evaluation set is currently not known to us. However, we will now present some results on either the development or progress set i.e., a collection of 16k recordings for online ranking.

For Task1, the language identification performance was measured on the development set with added noise. The data preparation protocol was explained in section 2. Equal error rate (EER) and C_{avg} are shown in Table 4.

For Task 3 we present the results for different WeNet models besides the one submitted for the sake of knowledge sharing. Table 3 presents the CER on the development of three different WeNet models of similar architecture. The first model is trained on 80 fbank features only. The second one is trained on fbank and pitch features (83 in total). While the last one corresponds to our multilingual WeNet submitted system trained on fbank and pitch features on top of noise augmented data. We observe that addition of the pitch features is beneficial for most of the OLR languages besides Russian and Tibetan; a result largely expected due to Russian's and Tibetan's atonal nature. Regarding the effect of the noise augmented data, we observe that certain languages show slight improvement while others slight degradation. Nevertheless, for the sake of robustness we decided to submit the noise augmented model, expecting more challenging recordings in the test set.

For Task 4, it is first worth discussing the performance of the individual models on the development set where the recordings' language is known i.e., LID is not required. Table 5 presents the CER scores of all models (monolingual and multilingual) used in our pipeline for all the OLR languages of the task. A number of interesting observations arise:

1. The performance varies depending on the language. This should not come as a surprise considering the diverse nature of the oriental languages. However, it suggests that a single solution cannot be applied across all languages at least on the context of this challenge.

[7] github.com/pytorch/fairseq/blob/main/examples/wav2vec/README.md

[8] dumps.wikimedia.org

[9] opus.nlpl.eu/OpenSubtitles-v1.php

[10] github.com/kpu/kenlm

	ru-ru	ct-cn	id-id	ja-jp	Kazak	ko-kr	Minnan	Shanghai	Sichuan	Tibet	Uyghu	vi-vn	zh-cn
fbank	31.99	24.90	14.60	35.02	13.12	34.33	68.50	40.23	28.12	29.11	10.48	9.70	28.65
fbank+pitch	33.76	17.96	13.00	23.00	11.94	27.44	62.99	32.26	20.16	31.55	9.71	7.89	19.84
fbank+pitch+noise	27.89	18.19	12.76	24.17	11.79	27.47	60.39	32.12	20.50	31.57	9.21	7.30	20.47

Table 3: *The CER performance of different WeNet models trained on different features and with or without noise augmented data. Our multilingual WeNet submission is trained on fbank+pitch features on noise augmented data (on top of the original).*

	EER (%)	C_{avg}
Pitch	0.64	0.0063
Non-pitch	0.52	0.0051
Combined	0.44	0.0044

Table 4: *The language identification performance measured on development set with noise added.*

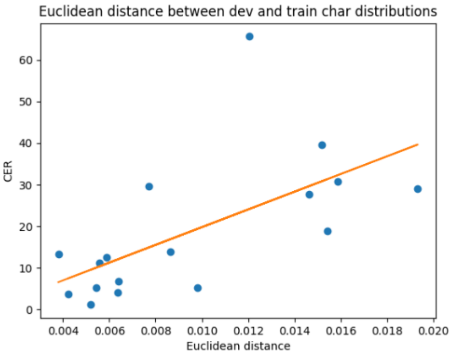

Figure 2: *Euclidean distance between train, dev character distributions versus the CER score using the finetuned XLSR-53 on various OLR languages. The larger the distance, the higher the CER.*

2. Chinese dialects i.e., ct-cn, zh-ch, Shanghai, Sichuan and Minnan, are some of the most problematic cases with the highest CER scores. The reason for this behavior can be attributed to two factors: 1) the large number of characters that constitute their alphabet and 2) the uneven distribution of characters between the training and development sets. This can been verified by plotting the Euclidean distance between the character distributions against the CER scores (see Figure 2). The figure clearly shows that imbalanced train, dev data are more likely to lead to high CER numbers.

3. WeNet multilingual is performing better on specific languages e.g. the Chinese dialects, than models specifically trained for those language without even using LID. This should not come as huge surprise; end-2-end models are known to be "data hungry" and training data comprising linguistically-related dialects can offer valuable information that monolingual models might miss.

Unfortunately due to time constraints, we could not produce the predictions on the development set using our complete cascaded pipeline. Nevertheless, on the progress set the MAFFT approach achieved our best results with a total CER of 27.5. In comparison, multilingual WeNet and wav2vec models (no LM) achieved 30.2 and 28.5 respectively.

7. Conclusions

We presented our system descriptions for Tasks 1,3 and 4 of the OLR2021 challenge. For task 1 we adopted the transformer model architecture by incorporating a language token in a sequence-to-sequence learning paradigm. For Task 3, we relied on a single end-to-end multilingual WeNet model. While for Task 4, we developed a cascaded approach by first performing language identification and then combining the predictions of wav2vec, WeNet monolingual and multilingual models. Through the course of the challenge we trained and employed numerous models and identified their strengths and weaknesses for each language and task. For our team the problem of LID and ASR on low-resource, oriental languages is still ongoing. As such, our future plans include researching and developing new solutions.

8. References

[1] D. Snyder, G. Chen, and D. Povey, "Musan: A music, speech, and noise corpus," *arXiv preprint arXiv:1510.08484*, 2015.

[2] 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.

[3] Z. Yao, D. Wu, X. Wang, B. Zhang, F. Yu, C. Yang, Z. Peng, X. Chen, L. Xie, and X. Lei, "Wenet: Production oriented streaming and non-streaming end-to-end speech recognition toolkit," in *Proc. Interspeech*, Brno, Czech Republic, 2021.

[4] 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.

[5] A. Baevski, H. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," *CoRR*, vol. abs/2006.11477, 2020. [Online]. Available: https://arxiv.org/abs/2006.11477

[6] M. Ott, S. Edunov, A. Baevski, A. Fan, S. Gross, N. Ng, D. Grangier, and M. Auli, "fairseq: A fast, extensible toolkit for sequence modeling," *arXiv preprint arXiv:1904.01038*, 2019.

[7] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *Proceedings of the 23rd international conference on Machine learning*, 2006, pp. 369–376.

[8] K. Katoh and D. M. Standley, "Mafft multiple sequence alignment software version 7: improvements in performance and usability," *Molecular biology and evolution*, vol. 30, no. 4, pp. 772–780, 2013.

Language	Model/Configuration	
id-id	Wav2Vec2: XLSR53 finetuned on OLR	5.07
	Wav2Vec2: XLSR53 finetuned on OLR + KenLM	4.50
	Wav2Vec2: XLSR53 finetuned on OLR and CommonVoice + KenLM	5.18
	WeNet Multilingual	12.76
	HuggingFace: indonesian-nlp/wav2vec2-large-xlsr-indonesian	9.07
vi-vn	Wav2Vec2: XLSR53 finetuned on OLR	5.05
	Wav2Vec2: XLSR53 finetuned on OLR + KenLM	4.47
	Wav2Vec2: XLSR53 finetuned on OLR and CommonVoice + KenLM	4.15
	WeNet Multilingual	7.30
	HuggingFace: nguyenvulebinh/wav2vec2-base-vietnamese-250h	3.40
ja-jp	Wav2Vec2: XLSR53 finetuned on OLR	21.60
	Wav2Vec2: XLSR53 finetuned on OLR + KenLM	18.76
	Wav2Vec2: XLSR53 finetuned on OLR and CommonVoice + KenLM	18.90
	WeNet Multilingual	24.17
	HuggingFace: jonatasgrosman/wav2vec2-large-xlsr-53-japanese	23.68
zh-cn	Wav2Vec2: XLSR53 finetuned on OLR	29.11
	Wav2Vec2: XLSR53 finetuned on OLR + KenLM	23.64
	WeNet Multilingual	20.47
	WeNet 20210421 unified conformer	12.0
	HuggingFace: jonatasgrosman/wav2vec2-large-xlsr-53-chinese-zh-cn	23.05
Kazak	Wav2Vec2: XLSR53 finetuned on OLR	7.26
	WeNet Multilingual	11.79
Shangai	Wav2Vec2: XLSR53 finetuned on OLR	39.69
	WeNet Multilingual	32.12
Sichuan	Wav2Vec2: XLSR53 finetuned on OLR	29.61
	WeNet Multilingual	20.50
Uighu	Wav2Vec2: XLSR53 finetuned on OLR	5.16
	WeNet Multilingual	9.21
ko-kr	Wav2Vec2: XLSR53 finetuned on OLR and OpenSLR	30.78
	Wav2Vec2: XLSR53 finetuned on OLR and OpenSLR + KenLM	25.02
	WeNet Multilingual	27.47
	HuggingFace: fleek/wav2vec-large-xlsr-korean	44.67
Minnan	Wav2Vec2: XLSR53 finetuned on OLR	65.64
	WeNet Multilingual	60.39
Tibet	Wav2Vec2: XLSR53 finetuned on OLR	13.32
	Wav2Vec2: XLSR53 finetuned on OLR + KenLM	12.11
	WeNet Multilingual	31.57
ct-cn	Wav2Vec2: XLSR53 finetuned on OLR	28.47
	WeNet Multilingual	18.19
	HuggingFace: ctl/wav2vec2-large-xlsr-cantonese	54.41
ru-ru	Wav2Vec2: XLSR53 finetuned on OLR and CommonVoice	13.80
	Wav2Vec2: XLSR53 finetuned on OLR and CommonVoice + KenLM	13.91
	WeNet Multilingual	27.89
	HuggingFace: jonatasgrosman/wav2vec2-large-xlsr-53-russian	17.60

Table 5: *The CER scores on the OLR development set for each individual model used in our pipeline for Task 4. WeNet multilingual was used for Task 3.*