TalTech System Description for the OLR 2021 Challenge

Tanel Alumäe, Kunnar Kukk

Tallinn University of Technology, Estonia

tanel.alumae@taltech.ee, kunnar.kukk@taltech.com

Abstract

This paper describes the spoken language identification systems developed in Tallinn University of Technology for the Oriental Language Recognition 2021 Challenge. We participated in two tracks: constrained and unconstrained language recognition. For the constrained track, we initialized the model by training a Conformer-based encoder-decoder model for multilingual speech recognition, using the provided training data that had transcripts available. The encoder part of the model was then finetuned for the language identification task. For the unconstrained task, we relied on pretrained models and external data: the pretrained XLSR-53 wav2vec2.0 model was finetuned on the VoxLingua107 corpus for the language recognition task, and finally finetuned on the provided training target language data, augmented with CommonVoice data. Our primary metric C_{avg} values on the Progess set are 0.0074 for the constrained task and 0.0119 for the unconstrained task.

Index Terms: spoken language recognition, OLR 2021 Challenge, VoxLingua107, transfer learning

1. Introduction

Spoken language identification (LID) is the task of automatically identifying the language of an utterance. Speech-based LID is used as a pre-processing step in several applications, such as automatic call routing, multilingual spoken translation and human-machine communication systems, multilingual speech transcription systems and spoken document retrieval. SLR is also often used in the area of intelligence and security.

In order to encourage the the research on multilingual phenomena and advance the development of language and dialect recognition technologies, the Oriental Language Recognition (OLR) Challenge has been organized annually since 2016 [1, 2, 3, 4, 5]. The sixth OLR challenge, denoted by OLR 2021 Challenge [6], included two language recognition tasks: constrained and unconstrained. The constrained task is a cross-domain closed-set identification task with 13 target languages. Only the data provided by the organizers can be used to build the system. The unconstrained task is a closed-set identification task with 17 languages. Here, utterances obtained from real-life environments. Any publicly available or proprietary data is allowed for system training and development.

The Tallinn University of Technology (TalTech) team particiapted in both language identification tasks. We relied on transfer learning: in the constrained task, we first trained a multilingual automatic speech recognition (ASR) model and finetuned it for language recognition, similarly as proposed in [7]. In the unconstrained task, we finetuned the multilingual XLSR-53 wav2vec2.0 model first on the VoxLingua107 dataset and then on the target language/dialect data.

2. Task 1: Constrained Language Identification

For the first task, our system was a combination of four models. One of the models was a Resnet-style model, trained on the provided training from scratch. The other three were finetuned from Conformer-based multilingual ASR models.

2.1. Resnet-style model

The Resnet-style mode is derived from the x-vector paradigm [8, 9], with several enhancements. During training, we apply on-the fly data augmentation using AugMix [10], by randomly distorting the training data using a mix of reverberation and noise augmentation. For frame-level feature extraction, we use the Resnet34 [11, 12] architecture where the basic convolutional blocks with residual connections are replaced with squeeze-and-excitation modules [13, 14]. The statistics pooling layer that maps frame-level features to segment level features is replaced in our model with a multi-head attention layer [15] that has been shown to provide superior performance [16, 17, 18, 19]. From among many variants of multi-head attention used in previous studies, we employ the one described in [16]: frame level representations are first mapped to N_{att} outputs ($N_{att} = 128$ in our model), using a 1×1 convolution and a ReLU nonlinearity; from this representation, each attention head (we used $N_{heads} = 5$ heads) computes it's own softmax-based weight distribution over the input utterance; finally, weighted mean and standard deviation are computed over the frame level features for each head, resulting in $N_{heads} \times 512 \times 2$ segment-level representations.

The structure of the embedding model is summarized in Table 1. The variables F and T refer to the number of filterbanks and the number of time frames in the utterance. In all experiments, we used $F = 30$.

The models are implemented in PyTorch [20] using a framework developed in our lab.

2.2. Conformer ASR model

The Conformer-based ASR model was trained on pooled provided training data that came with transcripts: we used the OLR 2016-2017 transcribed training data for the task 1 languages and OLR 2020 training data for the three Mandarin dialects. As a development set, we used the OLR 2020 test data. We applied a number of text normalization steps to the transcripts before training: for Cantonese, Mandarin, Mandarin dialects and Japanese, all whitespace symbols were deleted. For all languages, all punctuation symbols were deleted. For Kazakh, the Arabic script was transliterated into Cyrillic.

The ASR model uses a byte-pair encoding (BPE) vocabulary of 20000 units, shared over all languages. The ASR model is an encode-decoder based model that uses Conformer as an encoder and Transformer as a decoder. Some important hyperparameters of the model are listen in Table 2. The model was

Table 1: *The neural network architecture of the Resnet model. SE/res stands for squeeze-and-attention block and residual connections.*

Layer	*Spatial Size*	*#Channels*	*Kernel*
Input	$F \times T$	1	-
Frame level representations			
Pre-resnet	$F \times T$	64	7×7
Res-block 1	$F/2 \times T/2$	64	$3 \times \begin{bmatrix} 3 \times 3 \\ 3 \times 3 \\ SE/res \end{bmatrix}$
Res-block 2	$F/4 \times T/4$	128	$4 \times \begin{bmatrix} 3 \times 3 \\ 3 \times 3 \\ SE/res \end{bmatrix}$
Res-block 3	$F/8 \times T/8$	256	$6 \times \begin{bmatrix} 3 \times 3 \\ 3 \times 3 \\ SE/res \end{bmatrix}$
Res-block 4	$F/16 \times T/16$	512	$3 \times \begin{bmatrix} 3 \times 3 \\ 3 \times 3 \\ SE/res \end{bmatrix}$
Post-resnet	$1 \times T/16$	512	$F/16 \times 1$
Segment-level representations			
Pooling	1	$5 \times 512 \times 2$	Attent. stats
Embedding	1	512	Dense
FC	1	512	Dense
Output	1	#Languages	Softmax

trained on speed-perturbed data and SpecAugment [**?**] was applied during training. The number of training epochs was 40. After every epoch, the model's performance on development data was measured, and the final model was averaged over 10 best-performing models.

After training, the encoder part of the model was taken as the backbone of the language recognition model. The encoder's outputs were fed through a pooling layer. We experimented with different pooling methods: attentive statistics pooling [18], multi-head attention pooling [21], global multi-head attention pooling [22]. Two fully connected layers with the ReLu nonlinearity and BatchNorm were appended to the pooling layer. As a training criterion, we experimented with cross-entropy loss and additive angular margin loss [23]. The ASR encoder part of the model was trained together with the rest of the model, without any learning rate scaling. For some models, stochastic weight averaging (SWA) [24] was applied during the last 30% of training.

The language recognition model was trained on all available provided training data for the 13 target languages. Language embeddings were extracted from the first fully-connected layer after the pooling layer. The dimensionality of the embeddings was 512.

The Conformer-based model for ASR was trained using ESPnet [25]. For pooling operations and backend scoring (see below), ASV Subtools [26] was used.

2.3. Back-end modeling

For backend scoring, we used a multinomial regression model. Language embeddings were normalized and mean-centered based on the training data. Not LDA was used in this task. The logistic regression model was rebalanced in order to remove any bias due to the different amounts of training data per language.

We combine the scores of various systems using calibrated combination weights. For finding the model combination weights, we optimize the parameters of a linear model based on the log-likelihood cost metric (CLLR) on the development data, using L-BFGS as the optimizer. Our own Pytorch-based calibration implementation was used which is freely available[1].

Table 2: *Hyperparameters of the Conformer ASR model*

Conformer encoder	
Number of blocks	12
Linear dimensionality	2048
Dropout rate	0.1
Output size	256
Num. attention heads	8
Transformer decoder	
Linear dimensionality	2048
Number of blocks	6
Num. attention heads	4
Training	
CTC weight	0.3
Label smoothing	0.1

2.4. Results

Our results for Task 1 are listed in Table 3. It can be seen that transfer learning from an ASR model gives a massive boost to the system performance. Our best single model is based on the trained ASR Conformer encoder, used global multihead attention pooling, cross-entropy loss ans stochastic weight averaging. Fusion of four models gives only a slight improvement both on development and progress set. The fusion was also used in our official test set submission.

3. Task 2: Unconstrained Language Identification

3.1. Data

Task 2 allows the use of any publicly available or proprietary data for building the system. We relied a lot on the recently released VoxLingua107 dataset. The VoxLingua107 dataset [27][2] is a large-scale dataset for training spoken language identification models that work well on diverse real-life data.

VoxLingua107 was compiled from automatically scraped YouTube data. The data collection process is outlined on Figure 1. First, semi-random trigram search phrases were generated from the Wikipedia text corpus of the particular language. The search phrases were used to retrieve YouTube videos whose title or description matched the search phrase. Text-based language identification was used for filtering out the videos with the title and description likely not in the given language. Audio tracks of the videos were downsampled to 16 kHz. Speech activity detection and speaker diarization were applied for extracting segments from the videos that contain speech. Long speech segments were split into utterance-like subsegments of up to 20 seconds in length. Data-driven post-filtering was used to remove segments from the database that were likely not in the given language, increasing the proportion of correctly labeled

[1] https://github.com/alumae/sv_score_calibration

[2] Available at http://bark.phon.ioc.ee/voxlingua107/

Table 3: *Results on Task 1 with various systems and their combination.*

			Dev		*Progress*	
Backbone	*Pooling*	*Criterion*	C_{avg}	*EER*	C_{avg}	*EER*
Resnet	Attentive stats	CE	0.0515	6.90	0.0601	5.72
ASR Conformer	MHA	CE	0.0110	1.68	0.0101	1.15
ASR Conformer	GMHA	CE + SWA	0.0081	1.27	0.0080	0.91
ASR Conformer	MHA	AAM	0.0129	2.19	0.0133	1.68
Fusion			0.0078	1.20	0.0074	0.86

Table 4: *Results on Task 2 with various systems.*

	Dev		*Progress*	
Model	C_{avg}	*EER*	C_{avg}	*EER*
VoxLingua107 Resnet embeddings	0.053	5.15	0.078	7.98
VoxLingua107 Resnet embeddings, finetuned on training data	0.015	1.53	0.055	5.27
XLSR53, finetuned on VoxLingua107	0.017	1.76	0.016	1.69
XLSR53, finetuned on training data	0.003	0.29	0.044	3.78
XLSR53, finetuned on VoxLingua107, then on training data	0.006	0.78	0.012	0.93

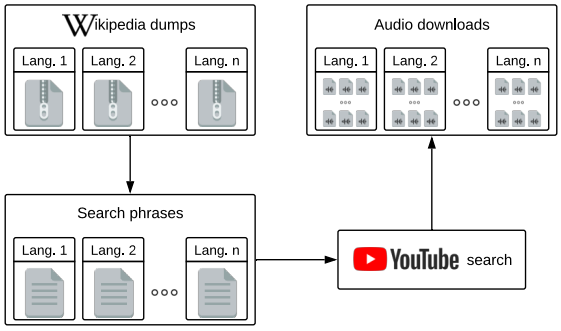

Figure 1: *Data collection process of VoxLingua107.*

Table 5: *Statistics about the VoxLingua107 dataset.*

Number of languages	107
Total number of videos	64110
Total number of hours	6682
Average number of hours per language	62
Average number of utterances per language	23709
Total amount of audio (uncompressed, in GB)	758

segments in the dataset to 98%, based on crowd-sourced verification. Some numerical facts about the VoxLingua107 training data are given in Table 5.

For final finetuning and training the backend logistic regression model, we used data audio in the provided OLR training data, and Mozilla CommonVoice data for the languages which were not present in the provided OLR training data: English, Hindi, Malay, Telugu and Thai. For languages with a lot of CommonVoice data, we limited the data to a random 15000 utterance subset. The development set was compiled from OLR2020 test data and CommonVoice data for the languages not covered by the OLR data.

3.2. Models

We experimented with using the XLSR-53 wav2vec2.0 model [28] as the backbone of our language embedding model. XLSR-53 is a large pretrained model trained on unlabeled multilingual data. The model is trained by jointly solving a contrastive task over masked latent speech representations and learning a quantization of the latents shared across languages. The model contains a convolutional feature encoder that maps raw audio to latent speech representations which are fed to a Transformer network that outputs context representations. XLSR-53 is pretrained on 56 000 hours of speech data from 53 languages.

We used XLSR-53 as follows: the outputs from the wav2vec2 model were fed through an attentive pooling layer, a fully connected layer with ReLU and BatchNorm, and the final output layer, corresponding to the languages of the training set. During training, the learning rate corresponding to the XLSR-53 model was set to 0.01 times lower than the base learning rate. We experimented with three finetuning scenarios: using VoxLingua107, using the training data of the 17 languages (OLR + CommonVoice), and using first VoxLingua107, followed by the training data for the 17 languages.

We also experimented with a Resnet model, trained on VoxLingua, and then finetuned on the OLR training data.

3.3. Back-end modeling

As in Task 1, we used a multinomial regression model for back-end scoring. Language embeddings were normalized and mean-centered based on the training data. The resulting embeddings were reduced to 50-dimensional vectors using LDA. No rebalancing of the model bias was used in this task, since it was found to slightly hurt the system performance on the progress set.

3.4. Results

The results of various systems are listed in Table 4. It can be seen that using XLSR-53 as the basis for finetuning models gives a big boost to the system performance, particularly on the progress set. Our internally compiled development set had quite

different performance trends, compared to the progress set. This is probably due to the fact that the development set contained clean dictated speech, whereas the progress set contained data "from the wild".

The best performing-model on the progress set was the XLSR-53 model, finetuned first on VoxLingua107 and then on the training data that covered all 17 target languages and dialects. This system was also used in our official test set submission.

4. Conclusion

This paper described the TalTech systems for the OLR 2021 Challenge. We focused on the spoken language identification tasks. For the constrained task, our best-performing single system was trained via transfer learning from the encoder part of a Conformer-Transformer based multilingual model. For the unconstrained task, we relied on two important external resources: the XLSR-53 pretrained multilingual wav2vec2 model, and the VoxLingua107 corpus. Our best performing model was trained via transfer learning from XLSR-53 using two finetuning steps: first using VoxLingua107, followed by target language training data.

5. References

[1] D. Wang, L. Li, D. Tang, and Q. Chen, "AP16-OL7: A multilingual database for oriental languages and a language recognition baseline," in *2016 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)*. IEEE, 2016, pp. 1–5.

[2] Z. Tang, D. Wang, Y. Chen, and Q. Chen, "AP17-OLR challenge: Data, plan, and baseline," in *2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*. IEEE, 2017, pp. 749–753.

[3] Z. Tang, D. Wang, and Q. Chen, "AP18-OLR challenge: Three tasks and their baselines," in *2018 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, 2018, pp. 596–600.

[4] Z. Tang, D. Wang, and L. Song, "AP19-OLR challenge: Three tasks and their baselines," in *2019 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, 2019, pp. 1917–1921.

[5] Z. Li, M. Zhao, Q. Hong, L. Li, Z. Tang, D. Wang, L. Song, and C. Yang, "AP20-OLR challenge: Three tasks and their baselines," in *2020 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, 2020, pp. 550–555.

[6] 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.

[7] D. Wang, S. Ye, X. Hu, S. Li, and X. Xu, "An end-to-end dialect identification system with transfer learning from a multilingual automatic speech recognition model," in *Interspeech*, 2021, pp. 3266–3270.

[8] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust DNN embeddings for speaker recognition," in *ICASSP*, 2018, pp. 5329–5333.

[9] D. Snyder, D. Garcia-Romero, A. McCree, G. Sell, D. Povey, and S. Khudanpur, "Spoken language recognition using x-vectors," in *Odyssey 2018 The Speaker and Language Recognition Workshop*, 2018.

[10] D. Hendrycks, N. Mu, E. D. Cubuk, B. Zoph, J. Gilmer, and B. Lakshminarayanan, "Augmix: A simple data processing method to improve robustness and uncertainty," *arXiv preprint arXiv:1912.02781*, 2019.

[11] W. Cai, J. Chen, and M. Li, "Exploring the encoding layer and loss function in end-to-end speaker and language recognition system," in *Odyssey 2018: The Speaker and Language Recognition Workshop*, 2018.

[12] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016, pp. 770–778.

[13] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *CVPR*, 2018, pp. 7132–7141.

[14] J. Zhou, T. Jiang, Z. Li, L. Li, and Q. Hong, "Deep speaker embedding extraction with channel-wise feature responses and additive supervision softmax loss function," in *Interspeech*, 2019, pp. 2883–2887.

[15] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[16] Y. Zhu, T. Ko, D. Snyder, B. Mak, and D. Povey, "Self-attentive speaker embeddings for text-independent speaker verification," in *Interspeech*, 2018.

[17] F. A. R. Rahman Chowdhury, Q. Wang, I. L. Moreno, and L. Wan, "Attention-based models for text-dependent speaker verification," in *ICASSP*. IEEE, 2018, pp. 5359–5363.

[18] K. Okabe, T. Koshinaka, and K. Shinoda, "Attentive statistics pooling for deep speaker embedding," in *Interspeech*, 2018.

[19] P. Safari and J. Hernando, "Self multi-head attention for speaker recognition," in *Interspeech*, 2019, pp. 4305–4309.

[20] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "PyTorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32*, 2019, pp. 8024–8035.

[21] M. India, P. Safari, and J. Hernando, "Self multi-head attention for speaker recognition," *arXiv preprint arXiv:1906.09890*, 2019.

[22] Z. Wang, K. Yao, X. Li, and S. Fang, "Multi-resolution multi-head attention in deep speaker embedding," in *ICASSP*, 2020, pp. 6464–6468.

[23] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "Arcface: Additive angular margin loss for deep face recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4690–4699.

[24] P. Izmailov, D. Podoprikhin, T. Garipov, D. Vetrov, and A. G. Wilson, "Averaging weights leads to wider optima and better generalization," *arXiv preprint arXiv:1803.05407*, 2018.

[25] S. Watanabe, T. Hori, S. Karita, T. Hayashi, J. Nishitoba, Y. Unno, N. Enrique Yalta Soplin, J. Heymann, M. Wiesner, N. Chen, A. Renduchintala, and T. Ochiai, "ESPnet: End-to-end speech processing toolkit," in *Proceedings of Interspeech*, 2018, pp. 2207–2211. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2018-1456

[26] F. Tong, M. Zhao, J. Zhou, H. Lu, Z. Li, L. Li, and Q. Hong, "ASV-Subtools: Open source toolkit for automatic speaker verification," in *ICASSP*. IEEE, 2021, pp. 6184–6188.

[27] J. Valk and T. Alumäe, "VoxLingua107: a dataset for spoken language recognition," in *Proc. IEEE SLT Workshop*, 2021.

[28] A. Conneau, A. Baevski, R. Collobert, A. Mohamed, and M. Auli, "Unsupervised cross-lingual representation learning for speech recognition," *arXiv preprint arXiv:2006.13979*, 2020.