Proceedings of the 13th Conference on Language Resources and Evaluation (LREC 2022), pages 7339–7343
Marseille, 20-25 June 2022

Deep learning-based end-to-end spoken language identification system for domain-mismatched scenario

Woo Hyun Kang, Jahangir Alam, Abderrahim Fathan
Computer Research Institute of Montreal
Montreal (Quebec) H3N 1M3, Canada
{woohyun.kang, jahangir.alam, abderrahim.fathan}@crim.ca

Abstract

Domain mismatch is a critical issue when it comes to spoken language identification. To overcome the domain mismatch problem, we have applied several architectures and deep learning strategies which have shown good results in cross-domain speaker verification tasks to spoken language identification. Our systems were evaluated on the Oriental Language Recognition (OLR) Challenge 2021 Task 1 dataset, which provides a set of cross-domain language identification trials. Among our experimented systems, the best performance was achieved by using the mel frequency cepstral coefficient (MFCC) and pitch features as input and training the ECAPA-TDNN system with a flow-based regularization technique, which resulted in a C_{avg} of 0.0631 on the OLR 2021 progress set.

Keywords: language identification, disentanglement, normalizing flow, OLR Challenge

1. Introduction

In recent years, various methods have been proposed utilizing deep learning architectures for language identification and have shown state-of-the-art performance when a large amount of in-domain training data is available [Gonzalez-Dominguez et al.2014, Snyder et al.2018a, N and Patil2021]. However, despite their success in well-matched conditions, the deep learning-based end-to-end identification systems are vulnerable to the performance degradation caused by mismatched conditions [Abdullah et al.2020].

In real life applications, numerous factors can contribute to the mismatches in language identification. Especially in a realistic scenario, the speech signals are likely to be not only collected from different devices, but from various environments [Ribas et al.2016]. Such mismatch is known to distort the speech signal differently, thus the language identification system should be robust against such adverse conditions in order to ensure reliable performance.

Recently, several attempts have been made to alleviate the domain-mismatch issue in the spoken language identification system. For example, in [Rangan et al.2020], the SpecAugment strategy [Park et al.2019] was adopted to make the language identification system robust against undesired spectral variability. In [Abdullah et al.2020], the authors employed the gradient reversal layer for training the network to be domain invariant. In [Shen et al.2017], a conditional generative adversarial network-based classification scheme was proposed to improve the generalization of the language identification system. Although these works have shown improvement in terms of performance, the domain-mismatch problem is still not extensively studied in the field of spoken language identification compared to other speech tasks (e.g., speaker verification, speech recognition).

In order to tackle these real-life issues, the OLR (Oriental Language Recognition) Challenge provides a standard benchmark for language identification systems on various mismatched conditions [Wang et al.2021]. Especially in Task 1, which restricts the choice of dataset used for training, the following problems should be considered:

- No in-domain data is provided for training or validating the language identification system.
- The primary performance metric is the C_{avg}, which considers the language-dependent false acceptance ratio and false rejection ratio. Therefore typical identification metrics, such as accuracy, will not reflect the systems C_{avg} performance.

More details about the OLR Challenge can be found in the evaluation plan [Wang et al.2021].

To solve these problems, we experimented with several architectures which have shown good results in other speech processing tasks, such as speaker verification. Moreover, we have adopted the flow-based embedding regularization (Flow-ER) [Kang et al.2021] strategy for disentangling the non-language information [Hansen and Hasan2015]. Among our experimented systems, the best performance was achieved via the mel frequency cepstral coefficient (MFCC) and pitch input ECAPA-TDNN system trained with the Flow-ER strategy.

2. Deep learning-based end-to-end language identification system

Classically, various attempts were done to use the phonotactic patterns for language identification [Li et al.2013]. The phonotactic approaches aim to distinguish the languages by comparing the frequency of occurences of certain sound sequences with that of the target languages. This is usually done by tokenizing

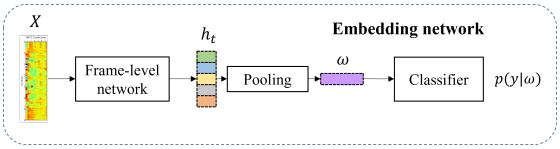


Figure 1: The deep learninag-based end-to-end language identification framework.

the speech signal using an automatic speech recognition system (ASR). However, due to the high computational load and the necessity of a transcribed dataset for training the ASR model, the language identification community has started to adopt methods used for speaker verification, which does not require any transcription [Dehak et al.2011, Castaldo et al.2007].
As the deep learning-based systems became the dominant apporach in the speaker verification field [Snyder et al.2018b, Snyder et al.2017, Desplanques et al.2020], these frameworks were naturally applied by the language identification community and have shown good performance [Gonzalez-Dominguez et al.2014, Snyder et al.2018a, N and Patil2021]. Especially when trained on a large dataset, it is shown that the deep learning-based system can outperform the conventional statistical method (e.g., i-vector) [Snyder et al.2018a].
As shown in Figure 1, the deep learning-based end-to-end language identification systems are composed of 3 modules: frame-level network, pooling layer, and a classifier. The frame-level network takes the acoustic feature extracted from the input speech and outputs a sequence of frame-level of representations. These deep representations are then aggregated into an utterance-level fixed-dimensional feature, which is also called an embedding vector, in the pooling layer. This embedding vector is then fed into a classifier network, which outputs the language probability.

3. System description

3.1. Backbone architecture

In our submissions, we adopted the following three architectures:

- ResNetSE34 [Chung et al.2020]: The first architecture is the Fast ResNet, which follows the same general structure as the original ResNet with 34 layers (ResNet-34) [He et al.2016] with squeeze-and-excitation [Hu et al.2018], but only uses one-quarter of the channels in each residual block to reduce computational cost.

- Hybrid [Alam et al.2021]: a CNN-LSTM-TDNN hybrid architecture with multi-level global-local statistics pooling, which has demonstrated good performance in various speaker verification tasks. The general framework for the Hybrid architecture is depicted in Figure 2.

- ECAPA-TDNN [Desplanques et al.2020]: an architecture that achieved state-of-the-art performance in text-independent speaker recognition. The ECAPA-TDNN uses squeeze-and-excitation as in the SE-ResNet, but also employs channel- and context-dependent statistics pooling and multi-layer aggregation.

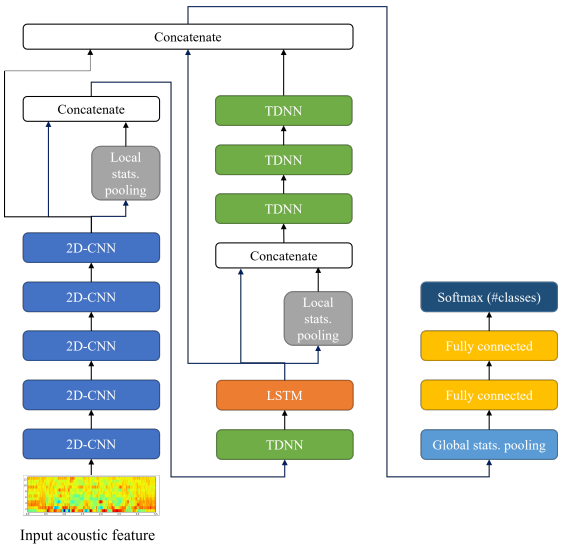


Figure 2: The general architecture of the hybrid embedding system.

For these architectures, we used two types of acoustic features:

- MFB: 40 dimensional mel-filterbank energy features,

- MFCC+Pitch: concatenation of 40 dimensional mel frequency cepstral coefficient (MFCC) and 3 dimensional pitch features.

Moreover, for the ResNetSE34 and ECAPA-TDNN systems, attentive statistical pooling (ASP) [Okabe et al.2018] layer was used to aggregate the frame-level representations, which was followed by a linear layer to obtain a 512-dimensional embedding vector.

3.2. Training objectives

3.2.1. Angular additive margin softmax (AAMSoftmax) objective

For training the ECAPA-TDNN-based systems, we used the angular additive margin softmax (AAMSoftmax) objective [Deng et al.2021]. The AAMSoftmax objective is formulated as follows:

$$L_{AAMSoftmax} = -\frac{1}{N}\sum_{i=1}^{N} log(\frac{e^{s(cos(\theta_{y_i,i}+m))}}{K_1}), \quad (1)$$

where $K_1 = e^{s(cos(\theta_{y_i,i}+m))} + \sum_{j=1,j\neq i}^{C} e^{scos\theta_{j,i}}$, N is the batch size, C is the number of classes, y_i corresponds to label index, $\theta_{j,i}$ represents the angle between the column vector of weight matrix W_j and the i-th embedding ω_i, where both W_j and ω_i are normalized.

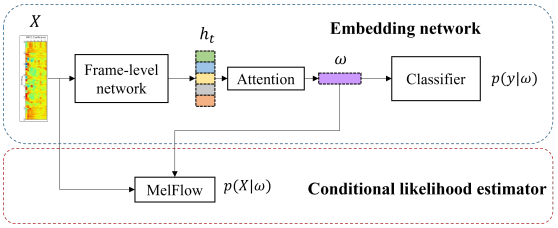


Figure 3: The flow-based embedding regularization (Flow-ER) framework.

The scale factor s is used to make sure the gradient is not too small during the training and m is the angular margin that encourages the similarity of correct classes to be greater than that of incorrect classes.

3.2.2. Flow-ER training strategy

In order to tackle the cross-domain problem, we have adopted the recently proposed flow-based embedding regularization strategy (Flow-ER) [Kang et al.2021]. In the Flow-ER framework, the embedding network is trained according to the information bottleneck scheme, where the mutual information between the embedding ω and the label y is maximized while the mutual information between ω and the input representation X is minimized. To accomplish this, we optimize the network with the following objective function:

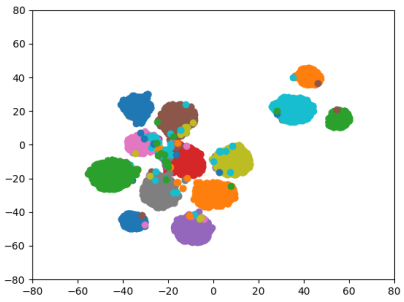
$$L_{IB} = -L_{xent} + \beta L_{redundancy}, \tag{2}$$

where β is a predefined coefficient, L_{xent} is the discriminative loss (e.g., AAMSoftmax), and $L_{redundancy}$ is the mutual information upperbound computed using the auxiliary MelFlow model as follows:

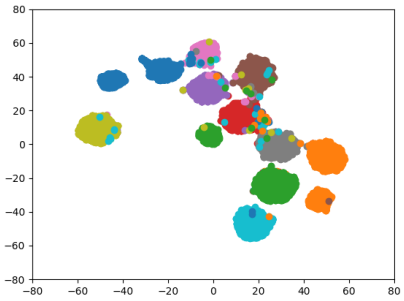
$$\begin{aligned} L_{redundancy} =& E_{p(X,\omega)}[\log p_X(X|\omega)] \\ & - E_{p(X)p(\omega)}[\log p_X(X|\omega)], \end{aligned} \tag{3}$$

where $\log p_X(X|\omega)$ is the conditional log-likelihood estimated using the MelFlow. The general Flow-ER framework is depicted in Figure 3.

The embedding network and the MelFlow network are trained in a competitive fashion, similar to the GAN training strategy. The Flow-ER training is done in a 2-stage process: embedding network update and MelFlow update. In the embedding network update phase, we freeze the MelFlow parameters and estimate the conditional likelihoods to compute $L_{redundacny}$. Then the embedding network and classification network parameters are updated through $L_{IB} = -L_{xent} + \beta L_{redundancy}$. In the MelFlow update phase, the embedding network parameters are frozen and the embeddings are extracted. Given the training data and their corresponding embeddings, the MelFlow is updated via likelihood maximization. Further information on the Flow-ER strategy can be found in [Kang et al.2021], and the MelFlow architecture can be found in [Kim et al.2020].



(a) System trained with no regularization.



(b) System trained with Flow-ER.

Figure 4: T-SNE plot of the language embeddings extracted from systems trained with and without Flow-ER. Different colors indicate distinct languages.

4. Results

4.0.1. Effects of the Flow-ER on the language embeddings

Figure 4 shows the T-SNE plot of the language embeddings extracted from systems trained with and without the proposed Flow-ER. From the embeddings extracted using the conventional method, it could be seen that some clusters are far away from each other even if they have the same class identity. Such variability may be attributed to the nuisance attributes, such as gender or speaker of the utterance. On the other hand, in the embeddings trained with the proposed Flow-ER, the clusters with the same class identity are relatively much closer to each other. From this observations, we could assume that the proposed Flow-ER can help the embeddings to have better discriminability by disentangling the nuisance attributes from them.

4.0.2. Language identification performance of systems with different configurations

Table 1 shows the equal error rate (EER) and average cost (C_{avg}) results of our submitted systems on the OLR 2021 Task 1 progress set. System 0 is the x-vector baseline result provided by the OLR Challenge organizers. As shown in the results, although the submitted systems generally performed well in terms of C_{avg}, the EER was very high in some systems (e.g., System 1, 2, 3). On the other hand, some systems with good EER

Table 1: Performance of the submitted systems on the OLR 2021 Progress set.

#	Architecture	Objective	Input	C_{avg}	EER [%]
0	Baseline			0.0826	9.038
1	Hybrid + LDA (20-dim.) + PLDA	Softmax	MFB	0.1364	47.220
2	Hybrid + LDA (30-dim.) + PLDA	Softmax	MFB	0.1360	47.290
3	Hybrid + LDA (20-dim.) + PLDA	Softmax	MFCC+pitch	0.1447	46.860
4	ResNetSE34	AAM	MFB	0.0951	10.180
5	ECAPA-TDNN	AAM	MFCC+pitch	0.0671	8.0940
6	ECAPA-TDNN	AAM + Flow-ER	MFB	0.0639	7.4370
7	**ECAPA-TDNN**	**AAM + Flow-ER**	**MFCC+pitch**	**0.0631**	**7.3340**
8	ECAPA-TDNN + LDA (20-dim.) + PLDA	AAM + Flow-ER	MFCC+pitch	0.4981	8.9400

have shown very bad performance in terms of C_{avg} (e.g., System 8). Such disparity between EER and C_{avg} is attributed to the different score statistics they consider. For example, while the EER considers the global false accept ratio (FAR) and false reject ratio (FRR), the C_{avg} computes the FAR and FRR conditioned on each target language. Therefore, training the language identification system similar to the speaker verification system can yield high EER performance, but will not guarantee good C_{avg} performance.

The ECAPA-TDNN-based systems (i.e., System 5, 6, 7) showed better performance than the Baseline, and the best performance was achieved by System 7. From these results, it could be seen that the Flow-ER strategy can improve the language identification performance in terms of both EER and C_{avg}. Especially System 7, which is trained with Flow-ER, outperformed the system trained without any embedding regularization (System 5) with a relative improvement of 5.96% in terms of C_{avg}. This indicates that the Flow-ER strategy can effectively minimize the non-language information from the language identification system.

Moreover, we could observe that in the ECAPA-TDNN-based systems, the MFCC+pitch acoustic feature (System 7) yielded better performance than the MFB feature (System 6), achieving a relative improvement of 1.25% in terms of C_{avg}. This may be attributed to the fact that prosodic cues, such as pitch, duration, or stress level differs greatly depending on the language. Therefore providing pitch features as input can give more discriminability to the language identification system.

5. Conclusions

In this paper, we have experimented with several deep learning-based models for language identification in domain mismatched scenario. The experimented systems were evaluated on the OLR Challenge Task 1, which consists of cross-domain trials, where the identification systems must be trained using a fixed training set. Moreover, this challenge does not provide any in-domain dataset for training or validating the systems. In order to overcome these problems, we experimented with several architectures that have shown good performance in the speaker verification task, such as ResNetSE34, ECAPA-TDNN and Hybrid systems. Moreover, in order to make the language identification system robust to domain mismatch, we have adopted the Flow-ER, which is a recently proposed regularization technique. Among the experimented methods, the best performance was achieved by the ECAPA-TDNN system which takes MFCC and pitch features as input and trained using AAMSoftmax and Flow-ER strategy. The best performing system achieved 0.0631 C_{avg} and 7.334% EER on the OLR 2021 progress set. From the results, we could notice a huge disparity between the C_{avg} and the EER metrics, which is due to the different statistics they consider.

In our future research, we plan to investigate new methods for training the system to jointly minimize the C_{avg} and the EER metrics. Furthermore, we will experiment with various fusion models to exploit the potential complementarity between different identification systems.

6. Acknowledgment

Authors wish to acknowledge Ministry of Economy and Innovation (MEI) of the Government of Quebec for the continued support.

7. Bibliographical References

Abdullah, B. M., Avgustinova, T., Möbius, B., and Klakow, D. (2020). Cross-domain adaptation of spoken language identification for related languages: The curious case of slavic languages. In Helen Meng, et al., editors, *Interspeech 2020, 21st Annual Conference of the International Speech Communication Association, Virtual Event, Shanghai, China, 25-29 October 2020*, pages 477–481. ISCA.

Alam, J., Fathan, A., and Kang, W. H. (2021). Text-independent speaker verification employing cnn-lstm-tdnn hybrid networks. In Alexey Karpov et al., editors, *Speech and Computer*, pages 1–13, Cham. Springer International Publishing.

Castaldo, F., Colibro, D., Dalmasso, E., Laface, P., and Vair, C. (2007). Compensation of nuisance factors for speaker and language recognition. *IEEE Transactions on Audio, Speech, and Language Processing*, 15(7):1969–1978.

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

Dehak, N., Torres-Carrasquillo, P. A., Reynolds, D. A., and Dehak, R. (2011). Language recognition via i-vectors and dimensionality reduction. In *Interspeech 2011*, pages 857–860.

Deng, J., Guo, J., Yang, J., Xue, N., Cotsia, I., and Zafeiriou, S. P. (2021). Arcface: Additive angular margin loss for deep face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1.

Desplanques, B., Thienpondt, J., and Demuynck, K. (2020). ECAPA-TDNN: emphasized channel attention, propagation and aggregation in TDNN based speaker verification. In Helen Meng, et al., editors, *Interspeech 2020*, pages 3830–3834. ISCA.

Gonzalez-Dominguez, J., Lopez-Moreno, I., and Sak, H. (2014). Automatic language identification using long short-term memory recurrent neural networks. In *Interspeech*.

Hansen, J. H. and Hasan, T. (2015). Speaker recognition by machines and humans: A tutorial review. *IEEE Signal Processing Magazine*, 32(6):74–99.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778.

Hu, J., Shen, L., and Sun, G. (2018). Squeeze-and-excitation networks. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7132–7141.

Kang, W. H., Alam, J., and Fathan, A. (2021). Robust speech representation learning via flow-based embedding regularization. *arXiv preprint*.

Kim, H., Lee, H., Kang, W. H., Kim, H. Y., and Kim, N. S. (2020). Robust front-end for multi-channel ASR using flow-based density estimation. *CoRR*, abs/2007.12903.

Li, H., Ma, B., and Lee, K. A. (2013). Spoken language recognition: From fundamentals to practice. *Proceedings of the IEEE*, 101(5):1136–1159.

N, K. D. and Patil, A. (2021). End-to-end language identification using multi-head self-attention and 1d convolutional neural networks.

Okabe, K., Koshinaka, T., and Shinoda, K. (2018). Attentive statistics pooling for deep speaker embedding. In *Interspeech*, pages 2252–2256.

Park, D. S., Chan, W., Zhang, Y., Chiu, C.-C., Zoph, B., Cubuk, E. D., and Le, Q. V. (2019). Specaugment: A simple data augmentation method for automatic speech recognition. *Interspeech 2019*, Sep.

Rangan, P., Teki, S., and Misra, H. (2020). Exploiting spectral augmentation for code-switched spoken language identification.

Ribas, D., Vincent, E., and Calvo, J. R. (2016). A study of speech distortion conditions in real scenarios for speech processing applications. In *2016 IEEE Spoken Language Technology Workshop (SLT)*, pages 13–20.

Shen, P., Lu, X., Li, S., and Kawai, H. (2017). Conditional Generative Adversarial Nets Classifier for Spoken Language Identification. In *Proc. Interspeech 2017*, pages 2814–2818.

Snyder, D., Garcia-Romero, D., Povey, D., and Khundanpur, S. (2017). Deep neural network embeddings for text-independent speaker verification. In *Interspeech*, pages 999–1003.

Snyder, D., Garcia-Romero, D., McCree, A., Sell, G., Povey, D., and Khudanpur, S. (2018a). Spoken Language Recognition using X-vectors. In *Proc. The Speaker and Language Recognition Workshop (Odyssey 2018)*, pages 105–111.

Snyder, D., Garcia-Romero, D., Sell, G., Povey, D., and Khudanpur, S. (2018b). X-vectors: robust dnn embeddings for speaker recognition. In *ICASSP*, pages 5329–5333.

Wang, B., Hu, W., Li, J., Zhi, Y., Li, Z., Hong, Q., Li, L., Wang, D., Song, L., and Yang, C. (2021). Olr 2021 challenge: Datasets, rules and baselines.