



Additive Phoneme-aware Margin Softmax Loss for Language Recognition

Zheng Li¹, Yan Liu¹, Lin Li¹, Qingyang Hong²

¹School of Electronic Science and Engineering, Xiamen University, China

²School of Informatics, Xiamen University, China

lilin@xmu.edu.cn, qyhong@xmu.edu.cn

Abstract

This paper proposes an additive phoneme-aware margin softmax (APM-Softmax) loss to train the multi-task learning network with phonetic information for language recognition. In additive margin softmax (AM-Softmax) loss, the margin is set as a constant during the entire training for all training samples, and that is a suboptimal method since the recognition difficulty varies in training samples. In additive angular margin softmax (AAM-Softmax) loss, the additional angular margin is set as a constant as well. In this paper, we propose an APM-Softmax loss for language recognition with phonetic multi-task learning, in which the additive phoneme-aware margin is automatically tuned for different training samples. More specifically, the margin of language recognition is adjusted according to the results of phoneme recognition. Experiments are reported on Oriental Language Recognition (OLR) datasets, and the proposed method improves AM-Softmax loss and AAM-Softmax loss in different language recognition testing conditions.

Index Terms: language recognition, oriental language recognition, margin loss, phonetic information

1. Introduction

Language recognition is a speech task to evaluate whether the target language is spoken in an auditory speech, and it is usually referred as Language Identification (LID), which is often set as the first process in a multilingual speech recognition system [1]. To encourage the improvement of LID technologies and to tackle the real challenge existing in LID tasks, with the support of oriental language databases, the OLR Challenge has been organized annually since 2016, attracting dozens of teams around the world [2, 3, 4, 5, 6].

LID and Automatic Speaker Verification (ASV) are two related speech tasks, sharing some key techniques. The Gaussian Mixture Model (GMM) based generative models with back-end processing have been dominant over the past decades [7, 8, 9]. In recent years, researchers have focused on exploring the usage of Deep Neural Network (DNN) in LID to improve performance in various testing conditions [10]. There were still many DNN based studies using i-vector, such as DNN i-vector [11], Bottleneck Feature (BNF) i-vector [12]. Later, end-to-end systems were also used to recognize languages [13]. More recently, motivated by the success of x-vector [14] in ASV, the DNN based language embedding has become a popular approach [15]. Based on the traditional x-vector structure, some deeper neural network structures were proposed, with the goal of learning a stronger language embedding extractor, such as C-NN based x-vector [16], extended TDNN based x-vector [17] and TDNN-F based x-vector [18].

The usage of phonetic information for ASV and LID tasks has also been studied. There is a variety of learning strategies of phonetic information for ASV [11, 19, 20, 21, 22]. However,

there is still a lot of research space for the usage of phonetic information in the field of LID. In [23], the senone-based Long Short-Term Memory (LSTM) Recurrent Neural Network (RN-N) was investigated to model long-range correlations in speech signals. In [24], a phonetic temporal model for LID was proposed. In our previous work [25], the phonetic multi-task learning framework was introduced into language recognition tasks.

Exploring a better loss function is always a hot topic in the community. The goal of a loss function is to enlarge inter-class distance while intra-class distance is reduced. Triplet loss [26, 27] minimizes the distance between feature pairs from the same classes and maximizes the distance between feature pairs from the different classes. Center loss was proposed in [28, 29]. Meanwhile, efforts have also been made to improve the Softmax loss with angular space, such as Angular Softmax (A-Softmax), Additive Margin Softmax (AM-Softmax) loss, and Additive Angular Margin Softmax (AAM-Softmax) loss [30, 31, 32, 33, 34, 35, 36].

In AM-Softmax loss [32], the additive margin is a hyperparameter, which is tuned before training and is a fixed constant for all training samples during the entire training period. While the angle between the feature vector and the center of its ground-truth class is scarcely the same for all training samples and the additive margin requires adaptive changes during training. And the additive angular margin is also a fixed hyperparameter in AAM-Softmax loss [36]. Further more, the usage of phonetic information in language-phoneme multi-task learning is inadequate, since it is only used as an auxiliary term in the final loss. Thus, it is more reasonable to use a dynamic margin or a dynamic angular margin based on phonetic information to improve AM-Softmax loss or AAM-Softmax loss. In this paper, we present an APM-Softmax for assisting language-phoneme multi-task learning. Specifically, the additive phoneme-aware margin for language classification is adjusted according to the cumulative highest posterior probability from phoneme samples of a segment speech. Experimental results on OLR datasets [6] show the effectiveness of the proposed APM-Softmax loss.

The rest of this paper is organized as follows. Section 2 introduces the related Softmax, A-Softmax, AM-Softmax, and AAM-Softmax loss. Section 3 presents the proposed additive phoneme-aware margin softmax loss. Section 4 shows the experimental setup and Section 5 gives experimental results and analysis. Finally, Section 6 concludes this paper.

2. Related Works

2.1. Margin Softmax Loss

The conventional Softmax loss is presented as:

$$L_S = -\frac{1}{N} \sum_{i=1}^N \log \frac{e^{W_{y_i}^T x_i + b_{y_i}}}{\sum_{j=1}^C e^{W_j^T x_i + b_j}} \quad (1)$$

where N is the number of training samples in a mini batch, C indicates the number of languages in the training set, x_i is the feature representation from the output of the last hidden layer of the i -th sample, y_i is the ground truth label of the i -th sample, W_j is the j -th column of the weights in the output layer, and b_j indicates the bias term.

The logit $W_{y_i}^T x_i + b_{y_i}$ can also be transformed to :

$$\|W_{y_i}\| \|x_i\| \cos(\Theta_{y_i}) + b_{y_i} \quad (2)$$

where Θ_{y_i} is the angle between W_{y_i} and x_i .

In A-Softmax [31], the weight vector is normalized and the bias term is discarded, in which $\|W_{y_i}\| = 1$ and $b_{y_i} = 0$, and the loss is presented as:

$$L_{AS} = -\frac{1}{N} \sum_{i=1}^N \log \frac{e^{\|x_i\| \phi(\Theta_{y_i})}}{e^{\|x_i\| \phi(\Theta_{y_i})} + \sum_{j=1; j \neq y_i}^C e^{\|x_i\| \cos(\Theta_j)}} \quad (3)$$

In the above equation, the original cosine angle $\cos(\Theta_{y_i})$ is replaced with $\phi(\Theta_{y_i})$:

$$\phi(\Theta_{y_i}) = (-1)^k \cos(m\Theta_{y_i}) - 2k \quad (4)$$

where $\Theta_{y_i} \in \left[\frac{k\pi}{m}, \frac{(k+1)\pi}{m}\right]$ and $k \in [0, m-1]$. The parameter m is used to control the size of the angular margin, which is a positive integer.

The AM-Softmax loss [32] further extends the angular based Softmax losses, which is written as:

$$L_{AMS} = -\frac{1}{N} \sum_{i=1}^N \log \frac{e^{s \cdot \phi(\Theta_{y_i})}}{e^{s \cdot \phi(\Theta_{y_i})} + \sum_{j=1; j \neq y_i}^C e^{s \cdot \cos(\Theta_j)}} \quad (5)$$

$$\phi(\Theta_{y_i}) = \cos(\Theta_{y_i}) - m \quad (6)$$

where the feature vector x_i in (3) is normalized to the unit, and is replaced with a hyper-parameter s .

The margin is larger with an additive penalty term m , which is also a hyper-parameter and is usually set as larger than zero.

More recently, AAM-Softmax loss [36] improves AM-Softmax [32] by adding the penalty on the angle directly, then Equation (5) and (6) are rewritten as:

$$L_{AAMS} = -\frac{1}{N} \sum_{i=1}^N \log \frac{e^{s \cdot \phi(\Theta_{y_i})}}{e^{s \cdot \phi(\Theta_{y_i})} + \sum_{j=1; j \neq y_i}^C e^{s \cdot \cos \Theta_j}} \quad (7)$$

$$\phi(\Theta_{y_i}) = \cos(\Theta_{y_i}) + m \quad (8)$$

where the angle Θ_{y_i} is directly penal by an additive penalty term m , which is a pre-set hyper-parameter as well.

2.2. Multi-task Learning

In this paper, we study the language-phoneme multi-task learning architecture [20, 21] based on the Extended Time Delay Neural Network (E-TDNN) [37], as shown in Figure 1. The language-phoneme multi-task learning framework contains three modules, including the shared E-TDNN feature encoder M_e at the frame-level, the phoneme classifier M_p at the frame-level and the language classifier M_c at the segment-level. Given a segment S of T frames $X = \{x_1, \dots, x_t, \dots, x_T\}$, the total

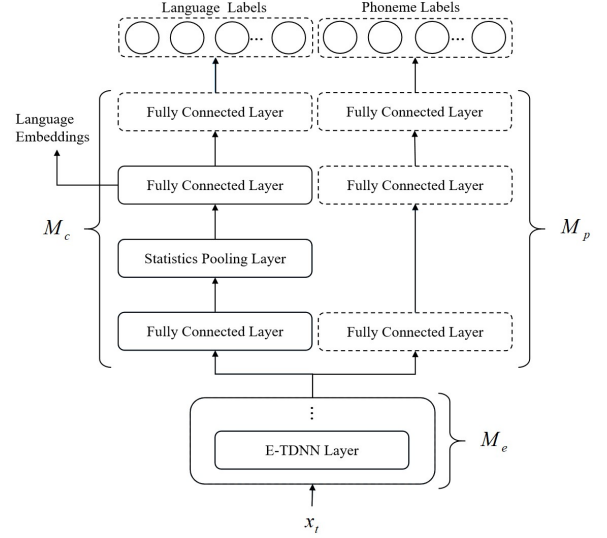


Figure 1: The language-phoneme multi-task learning structure. The dashed rectangles are removed after training.

loss of multi-task learning is composed of the language classification loss L_c and the phoneme classification loss L_p with an empirical control factor α , written as:

$$L_{total} = L_c + \alpha \cdot L_p \quad (9)$$

$$L_c = CE(M_c(M_e(X)), y^S) \quad (10)$$

$$L_p = \frac{1}{T} \sum_{t=1}^T CE(M_p(M_e(x_t)), y^P) \quad (11)$$

where $CE(A, B)$ indicates the Cross Entropy loss (Softmax loss) computed between the two distributions A and B . y^S denotes the segment-level language label, and y^P is the frame-level phoneme label.

After training, the dashed rectangles in Figure 1 are removed from the model, and the language embeddings are extracted from the penultimate layer in the language classifier M_c .

3. Additive Phoneme-aware Margin Softmax Loss

As mentioned in Section 2.1, in AM-Softmax loss or AAM-Softmax loss, the additive margin or additive angular margin is a constant during the entire training period for all training samples. However, the classification difficulty is barely the same among training samples, and it changes during the training. Thus, it is better to use a dynamic additive margin according to the training samples rather than using a fixed one. Further, the phonetic information is not fully utilized in language-phoneme multi-task learning mentioned in Section 2.2, since the phoneme classification sub-network is only used to constitute an auxiliary term in the total loss function.

This paper proposes an APM-Softmax loss for assisting language-phoneme multi-task learning, replacing the conventional Softmax loss in Equation (10). In brief, this approach introduces an adaptive phoneme-aware margin according to the

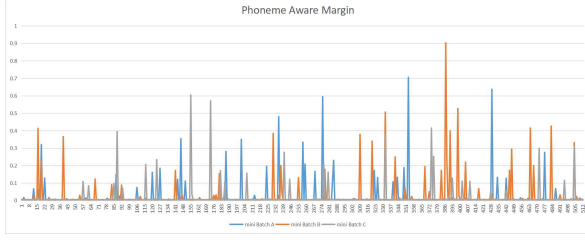


Figure 2: The additive phoneme-aware margins $\beta \cdot p$ in three random-selected mini batches, with the control factor $\beta = 10$. Each color indicates training samples from a mini batch.

cumulative highest posterior probabilities from phoneme samples in a speech segment. The proposed adaptive phoneme-aware margin is defined as:

$$P_i = m + \beta \cdot p_i \quad (12)$$

where m is a fixed basic additive (angular) margin, β is a control factor to control the strength of the phoneme-aware margin.

$$p_i = \frac{1}{T} \sum_{t=1}^T \max(\text{Softmax}(M_p(M_e(x_{t,i})))) \quad (13)$$

where T is the length of a chunk size in the training sample, $\max(\text{Softmax}(\cdot))$ is the operation for selecting the maximal posterior probability from the phoneme recognition sub-network's computation $M_p(M_e(\cdot))$ of the traing feature $x_{t,i}$.

Introducing the proposed adaptive phoneme-aware margin into AM-Softmax loss, the APM-Softmax is rewritten as:

$$L_{APMS} = -\frac{1}{N} \sum_{i=1}^N \log \frac{e^{s \cdot \rho(\Theta_{y_i})}}{e^{s \cdot \rho(\Theta_{y_i})} + \sum_{j=1; j \neq y_i}^C e^{s \cdot \cos(\Theta_j)}} \quad (14)$$

$$\rho(\Theta_{y_i}) = \cos(\Theta_{y_i}) - P_i \quad (15)$$

where the angular margin $\cos(\Theta_{y_i})$ is affected by the additive phoneme-aware margin P_i , which is changing among training samples i .

And into AAM-Softmax loss, Equation (15) is rewritten as:

$$\rho(\Theta_{y_i}) = \cos(\Theta_{y_i} + P_i) \quad (16)$$

where the angle Θ_{y_i} is directly impacted by the dynamic penalty term P_i .

As shown in Equation (13), in APM-Softmax, we use the average maximal phoneme posterior probability in a chunk of T features to indicate the degree of utterance level phoneme-aware, and it varies from training samples in every mini batch. It should be noted that, only the highest posterior probability is used for indicating the recognition difficulty of phonemes, rather than the posterior probability corresponding to its ground truth label.

Figure 2 shows the additive phoneme-aware margins $\beta \cdot p$ in three random-selected mini batches, with the control factor $\beta = 10$. In Figure 2, it can be observed that in each mini batch, the number of noteworthy additive phoneme-aware margins are minor. The number of those peaks are similar among different mini batches, indicating that training samples with high posterior probabilities are distributed evenly among mini batches, which is significant for the stationary iteration of training.

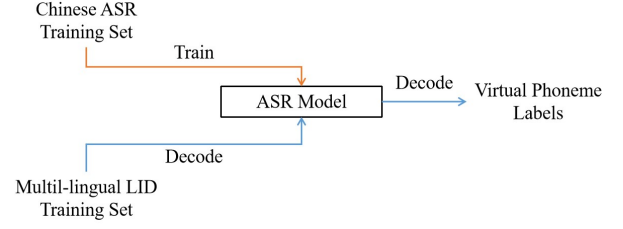


Figure 3: Preparation of virtual phoneme labels for language-phoneme multi-task learning.

4. Experimental Setup

4.1. Datasets

The language recognition database used in our experiments is the OLR dataset [2, 3, 4, 5, 6] for both training and testing.

In the OLR challenges, additional training materials were prohibited and participants were only permitted to use several specified datasets, including AP16-OL7, AP17-OL3, AP17-OLR-test, AP18-OLR-test, and THCHS-30 (plus the accompanying resources, such as dictionary and transcriptions) [38].

To better investigate of the proposed method, three kinds of test sets were used, which correspond to three particular LID conditions, including short utterance LID, cross-channel LID, and open-set dialect identification.

For short utterance LID, the AP18-OLR-test-short was used as the test set, with about 5.8 hours of speech. This data set is a close-set identification set, in which the languages of the utterances are as in the training set, but the test utterances are only one second long. For cross-channel LID, the AP19-OLR-test-channel was used, with about 8.9 hours of speech. This test data is recorded in different channels and is also a close-set identification set, including six known target languages: Cantonese, Indonesian, Japanese, Russian, Korean, and Vietnamese. For open-set dialect identification, the AP20-OLR-dev-task2 was used, with about 10.21 hours of utterances in total. The AP20-OLR-dev-task2 contains the dialect test subset of AP19-OLR-dev-task3, which contains three target dialects: Hokkien, Sichuanese, and Shanghaiese, and the test subset of AP19-OLR-test-task3, which contains three nontarget (interfering) languages: Catalan, Greek, and Telugu.

4.2. Phoneme Labels

For training a language-phoneme multi-task learning model, phoneme labels for the multi-lingual LID training set are necessary. Due to the constraint of the OLR Challenge rules, we only used THCHS-30 to get an ASR model, and to generate phoneme labels for the multi-lingual LID training set, as shown in Figure 3. Therefore, every language in the LID training set was decoded corresponding to Chinese phonemes, and this can be seem as virtual phonemes.

THCHS-30 [38] is an allowed-to-use data set in the OLR Challenges, which contains the lexicon and dictionary files for 30 hours of Chinese ASR training speech. Thus, based on the THCHS-30 training data, according to the Kaldi's recipe [39], a *tri4b* (tri-phone) GMM-based ASR model was trained, including 1,507 Probability Density Function Identifications (PDF-IDs). For the purpose of accurate phoneme alignment, we used the PDF-IDs in each frame to represent the corresponding phoneme labels. Finally, the virtual phoneme labels for the multi-lingual OLR training set were decoded from the trained Chinese ASR model.

Table 1: Experimental results on OLR dataset with the metric value C_{avg} .

No.	System	Loss	Margin (P)			Test Sets (Cavg)		
			m	β	p	Short-utt	Cross-channel	Open-set
1	x-vector	L_S	-			0.0540	0.2694	0.0851
2			-			0.0470	0.2779	0.0735
3	MT x-vector	L_{AMS}	0.2	-		0.0458	0.2769	0.1032
4			0.02	-		0.0467	0.2458	0.0792
5		L_{APMS}	0.2	10	0.002	0.0463	0.2497	0.0707
6		L_{AAMS}	0.2	-		0.0451	0.2731	0.0941
7			0.02	-		0.0441	0.274	0.0874
8		L_{APAMS}	0.2	10	0.002	0.0445	0.2590	0.0761

4.3. Training Details

Before training, we carried out data augmentation, including speed and volume perturbation, to increase the amount and diversity of the training data. The language-phoneme multi-task learning network was developed based on the OLR 2020 baseline E-TDNN x-vector, and we only added the phoneme classifier M_p as mention in Section 2.2. The acoustic features is 20-dimensional MFCC with 3-dimensional pitch, which has frame-length of 25ms, frame shifts of 10ms, and mean normalization over a sliding window of up to three seconds. Voice Activity Detection (VAD) was used to filter out non-speech frames. The back-end process was the same for all three test sets when the embeddings were extracted, as mentioned in the baseline system on OLR 2020 Challenge [6]. The front-end models were based on PyTorch [40, 41] optimized with Adam optimizer, with a mini-batch size of 512, and a chunk size of 100 frames in each segment, while the acoustic feature extraction and the back-end process were executed on Kaldi [39]. The hyper-parameter β of the phoneme-aware margin is empirically set to 10.

5. Results and Analysis

In this paper, the primary metric C_{avg} in [6] is chosen to evaluate the language recognition systems. Experimental results are given in Table 1. Thereinto, Softmax loss, AM-Softmax loss and AAM-Softmax loss are abbreviated as L_S , L_{AMS} and L_{AAMS} respectively. The AM-Softmax based APM-Softmax loss in Equation (15) is named as L_{APMS} , and AAM-Softmax based APM-Softmax loss in Equation (16) is as L_{APAMS} . In Table 1, x-vector indicates the baseline E-TDNN x-vector system and MT x-vector is for the multi-task learning E-TDNN x-vector system mentioned in 2.2.

Compared with the baseline x-vector in system No. 1, the improvement provided by language-phoneme multi-task learning is reported in system No. 2. Although the phoneme labels used to train the language-phoneme multi-task learning model were from a Chinese ASR model, which are named virtual phoneme labels in this paper, the introduction of phonetic information significantly improves the language recognition performance in Short-utterance task and Open-set task.

Moreover, it can be observed that introducing larger margins into language-phoneme multi-task learning x-vector models, as shown in system No. 3 and No. 6, improves the performance in Short-utterance task and Cross-channel task, while degrades that in Open-set task.

In order to further verify the effectiveness of the proposed APM-Softmax losses, based on the language-phoneme multi-task learning network, Table 1 shows the comparison of AM-

Softmax loss and AAM-Softmax loss with a little bit larger m , for compensating the additive phoneme margin $\beta \cdot p$ introduced into APM-Softmax losses. As shown in Figure 2, the additive phoneme margin p is varying within each mini batch during the training, thus, the p listed in Table 1 is the mean value from training samples of all mini batches, with $\beta = 10$. With a larger m , as reported in system No. 5, the performance of AM-Softmax loss is slightly improved in Cross-channel and Open-set tasks, but there is still a significant performance gap compared with APM-Softmax loss, in three LID tasks. And in AAM-Softmax, as in system No. 7, with a larger m , better results are achieved in Short-utterance task and Open-set task, while its contribution in those two tasks is much less than the proposed APM-Softmax, as in system No. 8. Moreover, the degradation on the Open-set task introduced by larger margin training strategies are significantly sloved with the dynamic additive phoneme-aware margin.

The results reinforce the assumption that dynamically setting the margin or angular margin according to the phonetic information for each training sample is more reasonable than using a constant margin or angular margin shared by all training samples.

6. Conclusions

In this paper, we propose an APM-Softmax loss for improving language-phoneme multi-task learning. In the presented APM-Softmax loss, the additive margin changes according to the phoneme recognition results of each language training samples during training, which improves the fixed margin AM-Softmax loss, and the fixed angular margin AAM-Softmax loss. The phoneme-aware method adopts an adaptive margin according to the phonetic information. We investigated the proposed method with the E-TDNN based multi-task learning architecture on OLR dataset. Experimental results show that our proposal improves the AM-Softmax loss and the AAM-Softmax loss based systems, under three different language recognition conditions.

In the future work, more usages of the phonetic information for language recognition will be investigated. Moreover, the proposed additive phoneme-aware margin loss is also suitable for other speech classification tasks, such as speaker recognition task, thus, more studies will be given in the applications of APM-Softmax loss.

7. Acknowledgements

This work is supported by the National Natural Science Foundation of China (Grant No. 61876160, Grant No. 62001405).

8. References

- [1] F. Castaldo, D. Colibro, S. Cumani, E. Dalmasso, P. Laface, and C. Vair, "Loquendo-Politecnico di Torino system for the 2009 NIST language recognition evaluation," in *ICASSP*, 2010, pp. 5002–5005.
- [2] D. Wang, L. Li, D. Tang, and Q. Chen, "AP16-OL7: A multilingual database for oriental languages and a language recognition baseline," pp. 1–5, 2016.
- [3] Z. Tang, D. Wang, Y. Chen, and Q. Chen, "AP17-OLR challenge: Data, plan, and baseline," in *APSIPA ASC*, 2017, pp. 749–753.
- [4] Z. Tang, D. Wang, and Q. Chen, "AP18-OLR challenge: Three tasks and their baselines," in *APSIPA ASC*, 2018.
- [5] Z. Tang, D. Wang, and L. Song, "AP19-OLR challenge: Three tasks and their baselines," in *APSIPA ASC*, 2019.
- [6] 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 *APSIPA ASC*. IEEE, 2020, pp. 550–555.
- [7] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, pp. 788–798, 2011.
- [8] N. Dehak, P. Torres-Carrasquillo, D. Reynolds, and R. Dehak, "Language recognition via i-vectors and dimensionality reduction," in *Interspeech*, 01 2011, pp. 857–860.
- [9] D. Martinez, O. Plchot, L. Burget, O. Glembek, and P. Matejka, "Language recognition in ivectors space," in *Interspeech*, 01 2011, pp. 861–864.
- [10] "Automatic language identification using deep neural networks," in *IEEE International Conference on Acoustics*, 2014.
- [11] Y. Lei, N. Scheffer, L. Ferrer, and M. McLaren, "A novel scheme for speaker recognition using a phonetically-aware deep neural network," in *ICASSP*, 2014, pp. 1695–1699.
- [12] F. Richardson, D. Reynolds, and N. Dehak, "Deep neural network approaches to speaker and language recognition," *IEEE Signal Processing Letters*, vol. 22, no. 10, pp. 1671–1675, 2015.
- [13] I. Lopez-Moreno, J. Gonzalez-Dominguez, O. Plchot, D. Martinez, J. Gonzalez-Rodriguez, and P. Moreno, "Automatic language identification using deep neural networks," in *ICASSP*, 2014, pp. 5337–5341.
- [14] D. Snyder, D. Garcia-Romero, A. McCree, G. Sell, and S. Khudanpur, "Spoken language recognition using x-vectors," 06 2018.
- [15] S. Sadjadi, T. Kheyrkhan, A. Tong, C. Greenberg, D. Reynolds, E. Singer, L. Mason, and J. Hernandez-Cordero, "The 2017 NIST language recognition evaluation," 06 2018, pp. 82–89.
- [16] H. Zeinali, L. Burget, and J. Cernocky, "Convolutional neural networks and x-vector embedding for DCASE2018 acoustic scene classification challenge," *arXiv preprint arXiv:1810.04273*, 2018.
- [17] D. Garcia-Romero, D. Snyder, G. Sell, A. McCree, D. Povey, and S. Khudanpur, "X-vector DNN refinement with full-length recordings for speaker recognition," in *Interspeech*, 2019, pp. 1493–1496.
- [18] J. Villalba, N. Chen, D. Snyder, D. Garcia-Romero, A. McCree, G. Sell, J. Borgstrom, F. Richardson, S. Shon, F. Grondin *et al.*, "State-of-the-art speaker recognition for telephone and video speech: the JHU-MIT Submission for NIST SRE18," *Interspeech*, pp. 1488–1492, 2019.
- [19] P. Kenny, V. Gupta, T. Stafylakis, P. Ouellet, and J. Alam, "Deep neural networks for extracting baum-welch statistics for speaker recognition," in *Odyssey*, 2014.
- [20] S. Wang, J. Rohdin, L. Burget, O. Plchot, Y. Qian, K. Yu, and J. ernock, "On the usage of phonetic information for text-independent speaker embedding extraction," in *Interspeech*, 09 2019, pp. 1148–1152.
- [21] Y. Liu, L. He, J. Liu, and M. Johnson, "Speaker embedding extraction with phonetic information," in *Interspeech*, 09 2018, pp. 2247–2251.
- [22] Y. Yang, S. Wang, X. Gong, Y. Qian, and K. Yu, "Text adaptation for speaker verification with speaker-text factorized embeddings," in *ICASSP*, 2020, pp. 6454–6458.
- [23] Y. Tian, L. He, Y. Liu, and J. Liu, "Investigation of senone-based long-short term memory RNNs for spoken language recognition," in *Odyssey*, 2016, pp. 89–93.
- [24] Z. Tang, D. Wang, Y. Chen, L. Li, and A. Abel, "Phonetic temporal neural model for language identification," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 1, pp. 134–144, 2018.
- [25] L. Li, Z. Li, Y. Liu, and Q. Hong, "Deep joint learning for language recognition," *Neural Networks*, vol. 141, pp. 72–86, 2021.
- [26] C. Li, X. Ma, B. Jiang, X. Li, X. Zhang, X. Liu, Y. Cao, A. Kannan, and Z. Zhu, "Deep speaker: an end-to-end neural speaker embedding system," *arXiv preprint arXiv:1705.02304*, vol. 650, 2017.
- [27] C. Zhang and K. Koishida, "End-to-end text-independent speaker verification with triplet loss on short utterances," in *Interspeech*, 2017, pp. 1487–1491.
- [28] N. Li, D. Tuo, D. Su, Z. Li, D. Yu, and A. Tencent, "Deep discriminative embeddings for duration robust speaker verification," in *Interspeech*, 2018, pp. 2262–2266.
- [29] S. Yadav and A. Rai, "Learning discriminative features for speaker identification and verification," in *Interspeech*, 2018, pp. 2237–2241.
- [30] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song, "Sphereface: Deep hypersphere embedding for face recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 212–220.
- [31] Y. Li, F. Gao, Z. Ou, and J. Sun, "Angular softmax loss for end-to-end speaker verification," in *2018 11th International Symposium on Chinese Spoken Language Processing (ISCSLP)*. IEEE, 2018, pp. 190–194.
- [32] F. Wang, J. Cheng, W. Liu, and H. Liu, "Additive margin softmax for face verification," *IEEE Signal Processing Letters*, vol. 25, no. 7, pp. 926–930, 2018.
- [33] M. Hajibabaei and D. Dai, "Unified hypersphere embedding for speaker recognition," *arXiv preprint arXiv:1807.08312*, 2018.
- [34] Y.-Q. Yu, L. Fan, and W.-J. Li, "Ensemble additive margin softmax for speaker verification," in *ICASSP*. IEEE, 2019, pp. 6046–6050.
- [35] Y. Liu, L. He, and J. Liu, "Large margin softmax loss for speaker verification," *arXiv preprint arXiv:1904.03479*, 2019.
- [36] 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.
- [37] J. Villalba, N. Chen, D. Snyder, D. Garcia-Romero, A. McCree, G. Sell, B. Borgstrom, F. Richardson, S. Shon, F. Grondin, R. Dehak, P. Garcia, D. Povey, P. Torres-Carrasquillo, S. Khudanpur, and N. Dehak, "State-of-the-art speaker recognition for telephone and video speech: The JHU-MIT submission for NIST SRE18," in *Interspeech*, 09 2019, pp. 1488–1492.
- [38] D. Wang and X. Zhang, "THCHS-30: A free Chinese speech corpus," *ArXiv*, vol. abs/1512.01882, 2015.
- [39] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesel, "The Kaldi speech recognition toolkit," *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*, 01 2011.
- [40] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in PyTorch," in *NIPS-W*, 2017.
- [41] 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 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 6184–6188.