Journal of Computer Science and Cybernetics, V.40, N.3 (2024), 287–298
DOI no 10.15625/1813-9663/18320

LANGUAGE-ADVERSARIAL TRAINING FOR INDIC MULTILINGUAL SPEAKER VERIFICATION

HOANG LONG VU, NGUYEN VAN HUY, NGO THI THU HUYEN, PHAM VIET THANH*

Hanoi University of Science and Technology,
1 Dai Co Viet Street, Hai Ba Trung District, Ha Noi, Viet Nam

Abstract. Speaker verification now reports a reasonable level of accuracy in its applications in voice-based biometric systems. Recent research on deep neural networks and predicting speaker identity based on speaker embeddings have gained remarkable success. However, results are limited when it comes to verifying multilingual speakers. In this paper, we propose an ensemble system submitted to the I-MSV Challenge 2022. The system is built upon the ECAPA and RawNet model with additional adversarial training layers. Probabilistic Linear Discriminant Analysis back-end scoring and Large Margin Cosine Loss are implemented to further obtain more discriminative features. Experimental results show that on the Constraint Private Test set of the task, our proposed model achieved remarkable results, ranked third with an Equal Error Rate (EER) of 2.9734%.

Keywords. Speaker verification, adversarial training, multilingual.

1. INTRODUCTION

Speaker verification (SV) is the task of verifying the identity of a person from the characteristics of the voice signal. The verification is conducted between a test utterance spoken at test time, and an enrolment utterance. A robust SV system is expected to perform effectively without depending on the variants in emotion and the language of a speaker. [1] showed that there is significant degradations in case of mismatches between test and enrolment speeches. Therefore, despite the active research and remarkable achievements in SV technologies, the development concerning multilingual conversation is still limited. Former works focused on frameworks based on Gaussian Mixture Model (GMM) to capture acoustic feature distribution and spectral shapes to tackle mismatch in speaking languages [1]. The authors in this work also proposed a major challenge of collecting adequate data for preparing a multilingual speech corpus.

In India, most of the speakers are multilingual and the speaking style varies across geographical regions. The COCOSDA INDIC-Multilingual Speaker Verification (I-MSV) Challenge 2022 comprises speech data from 13 Indian languages, collected using different sensors to make the SV system robust to language and sensor variations between enrollment and testing.

*Corresponding author.
E-mail addresses: longvu200502@gmail.com (H.L. Vu); huydsai02@gmail.com (N.V. Huy); huyenthu432002@gmail.com (N.T.T. Huyen); thanh.pv.ds@gmail.com (P.V. Thanh)

Current state-of-the-art results for speaker verification are achieved from active research in deep neural networks. Improved upon the original Time Delay Neural Network (TDNN) [2] architecture, the ECAPA [3] model allows the extraction of speaker embedding vectors from the first fully connected layer after a statistical pooling layer. The cosine similarity or Probabilistic Linear Discriminant Analysis (PLDA) [4] is often adopted as a back-end scoring model to handle the channel mismatch between enrollment speakers and the evaluation speech. RawNet [5] - the first end-to-end model in speaker verification using raw waveforms extracts the frame-level embeddings using residual blocks with Convolutional Neural Network (CNN), then aggregates features into utterance level using Long Short-term Memory (LSTM). This RawNet model is proven to have comparable results to the TDNN architecture [5].

In this paper, we propose an SV system submitted to the I-MSV Challenge 2022. The system utilizes the ECAPA and RawNet model with PLDA back-end scoring. To further reduce the variability of intra-class and enlarge the inter-class distance to obtain more discriminative features, the Large Margin Cosine Loss (LMCL) [6] function is applied as the objective loss function for the models. Adversarial training [7] is also applied to reduce the language effect of the utterances. Moreover, score normalization is necessary to set the same single detection threshold for the scores obtained from the different speaker models, in other words, to produce well calibrated speaker verification scores in SV systems. Our contribution from this work is a experimental proof that adversarial training and Large Margin Cosine Loss managed to improve the performance of SV systems.

The remaining of this paper is organized as follows. The related works are described in Section 2, followed by the methodologies described in Section 3. In Section 4, we discuss the experimental setup and show the results of the systems in Section 5. Finally, the conclusions are drawn in Section 6.

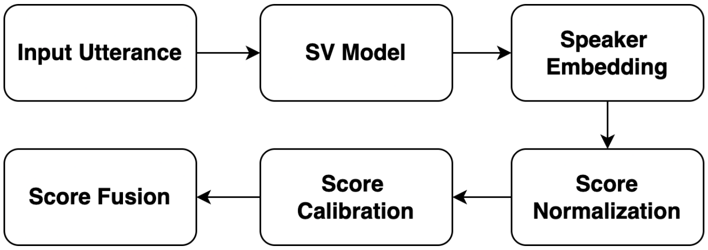

Figure 1: The overall pipeline

2. RELATED WORKS

Previous works have shown the efficiency of verifying speaker identity based on the extracted speaker embeddings in [3, 5]. Furthermore, the PLDA back-end scoring model also provides consistent performance improvements in speaker verification evaluations, as shown in [4, 8].

Multilingual speaker verification is still a challenging research area. One potential challenge in these tasks is to gather enough training data for different languages. To tackle this resource scarcity, different approaches such as semi-supervised training, unsupervised

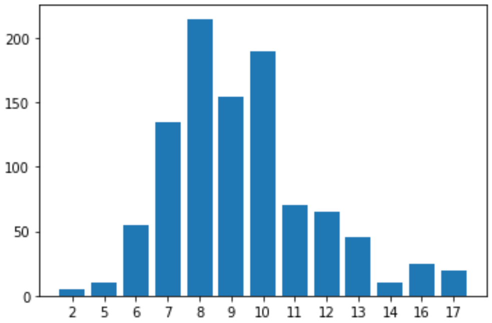

Figure 2: The duration distribution of original training utterances

training, and multitasking learning have been proposed [9, 10]. It is proved that multilingualism allows shared training data from multiple languages to cover a wider acoustic context. Recently, the Domain Adversarial Network (DAN) was designed to reduce mismatches in domains by adversarially learning domain-invariant features [11]. Adversarial training for multilingual tasks is shown to reduce the language effect in utterances. The adversarial models can learn to generate embeddings that do not contain language-specific information, resulting in improved overall performance for multilingual speaker verification problems [7, 11, 12].

The LMCL and Additive Angular Margin (AAM) Loss, which were traditionally more common in deep face recognition tasks, have also been implemented with promising results in speaker verification tasks [13, 14].

3. METHODOLOGY

Our proposed method undergoes three main stages: After obtaining speaker embeddings from the SV models, we normalize the scores and calibrate those outputs to prepare for the final fusion. The pipeline is illustrated in Figure 1.

3.1. Data pre-processing

The training data provided by the competition organizer comprises 100 utterances from 50 different speakers. The utterance length distribution is illustrated in Figure 2. To match the duration of test data (10 to 60 seconds), we split every training utterance into smaller intervals of approximately the same duration range as the test utterances. After splitting, we obtained over 16, 000 utterances of length from 10 to 60 seconds, without any overlap between utterances.

3.2. Models

3.2.1. ECAPA-TDNN

Modern speaker embeddings are extracted from deep neural models trained to discriminate speaker identities from a large pool of speakers.

A temporal statistics pooling layer is used to map the variable length input to a fixed-length representation. After training, the fixed-length speaker embeddings are extracted from the activations of the penultimate layer in the network. The pooling layer uses a channel- and context-dependent attention mechanism, which allows the network to attend different frames per channel. 1-dimensional SqueezeExcitation blocks rescale the channels of the intermediate frame-level feature maps to insert global context information in the locally operating convolutional blocks. Next, the integration of 1-dimensional Res2-blocks improves performance while simultaneously reducing the total parameter count by hierarchically using grouped convolutions. Finally, Multi-layer Feature Aggregation merges complementary information before the statistics pooling by concatenating the final frame-level feature map with intermediate feature maps of preceding layers.

3.2.2. RawNet3

RawNet [5] is a neural speaker embedding extractor that inputs raw waveforms directly without preprocessing techniques and outputs speaker embeddings designed for speaker verification.

The underlying assumption behind using a DNN is that speaker embeddings extracted directly from raw waveforms by replacing an acoustic feature extraction with more hidden layers are expected to yield more discriminative representations as the amount of available data increases. RawNet adopts a convolutional neural network-gated recurrent unit (CNN-GRU) architecture, in which the first CNN layer has a stride size identical to the filter length. The front CNN layers comprise residual blocks followed by a max-pooling layer and extract frame-level representations. Then a Gated Recurrent Unit (GRU) layer aggregates frame-level features into an utterance-level representation, which is the final timestep of the GRU's output. The GRU layer is then connected to a fully connected layer, where its output is used as the speaker embedding. Finally, the output layer receives a speaker embedding and performs identification in the training phase.

3.3. Back-end scoring

Speaker verification can be accomplished by calculating the similarity between the two speaker embeddings of the enrollment and test speech, which can be measured by a simple cosine distance. Alternatively, we can opt for a more sophisticated, supervised back-end model like PLDA which involves the explicit use of between and within covariance matrices.

3.3.1. Cosine similarity

Initially, we used cosine similarity to evaluate how close two speaker embeddings and this serves as the simplest back-end. When this method is applied to speaker verification, the cosine of the angle between the enrollment (ϕ_e) and test (ϕ_t) embeddings is used as the decision score

$$s(\phi_e, \phi_t) = \frac{\langle \phi_e, \phi_t \rangle}{\parallel \phi_e \parallel \parallel \phi_t \parallel}. \tag{1}$$

This scoring technique just involves the inter-product of two speaker embedding vectors that need to be scored. Therefore, the results were just quite good on the public test (shown in Table 1). We tried using Probabilistic linear discriminant analysis – another back-end scoring, and the results improved significantly.

3.3.2. Probabilistic linear discriminant analysis (PLDA)

Theoretically, PLDA extends the traditional linear discriminant analysis in a probabilistic way. For our models, we used SpeechBrain [15] - an open-source and all-in-one speech toolkit to train PLDA and compute PLDA score. According to SpeechBrain, PLDA modeling bases on i-vector [16] and supervector space for speaker verification. An i-vector that is a low-dimensional vector contains both speaker and channel information acquired from a speech segment [4]. Dimension reduction occurs twice when PLDA is applied to an i-vector: once during the i-vector extraction process and once again during the PLDA model [4].

To avoid losing important information, we keep the i-vector at its full dimensionality when using PLDA for modeling and scoring [4]. PLDA is a technique that takes i-vectors as input and is trained on a training set. The results we obtained using PLDA back-end scoring on the public test set (shown in Table 1) are much better than those obtained using cosine similarity. Therefore, we decided to use PLDA back-end scoring for the private test set.

3.4. Loss function

3.4.1. Additive angular margin loss

ArcFace, or AAM Loss [17], is a loss function traditionally used in face recognition tasks. The ArcFace loss function is an improved version of the Softmax loss function, which can directly maximize the classification boundary in the angular space and improve the classification accuracy.

The formula for AAM Loss is defined as follows

$$L_{AAM} = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{s\left(\cos\left(\theta_{y_i}+m\right)\right)}}{e^{s\left(\cos\left(\theta_{y_i}+m\right)\right)}+\sum_{j=1,j\neq y_i}^{n} e^{s\cos\theta_j}}. \tag{2}$$

The AAM Loss function employs an additive angular margin penalty m between x_i and the weights W_{y_i} to simultaneously enhance the intra-class compactness and inter-class discrepancy.

However, the margin of AAM Softmax is not consistent with all values of θ: the margin becomes smaller as θ reduces and vanishes completely when θ shrinks to 0.

3.4.2. Large margin cosine loss

The LMCL [6] defines the decision margin in cosine space rather than in the angle space and is formulated as

$$L_{LMC} = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{s}(\cos(\theta_{y_i}, i))}{e^{s(\cos(\theta_{y_i},i)-m)}+\sum_{j\neq y_i} e^{s}\cos(\theta_j, i)}. \tag{3}$$

Subject to,

$$W = \frac{W^*}{\| W^* \|},\ x = \frac{x^*}{\| x^* \|},\ \cos(\theta_j, i) = W_j^T x_i, \tag{4}$$

where N is the number of training examples, x_i is the feature vector with corresponding label y_i, W_j is the weight vector, and θ_j is the angle between W_j and x_i.

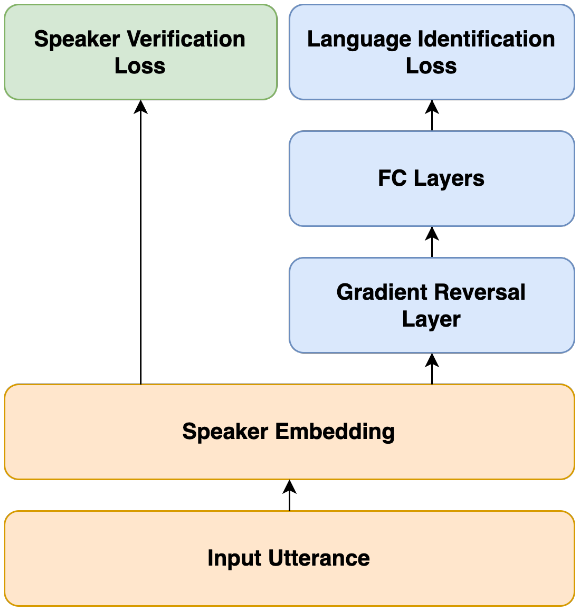

Figure 3: Adversarial training architecture

3.5. Adversarial training

Adversarial training is a technique commonly used in domain adaptation. Rather than making the model focuses on the domain information, this technique forces the model to learn domain-invariant features, therefore improving the generalization across all domains. In our proposed method, domain can be considered the language aspect.

In a multitask learning framework, an adversarial head is jointly trained with a classification task to classify training samples into different domains. By reversing the gradients, the lower-level layers of the network try to minimize the domain difference and learn domain-invariant features. DAN have found utility in the field of ASR by enhancing the robustness and flexibility of systems to address an array of complex real-world scenarios [12].

In this work, we propose to use domain adversarial training to promote data sharing between different languages.

Given supervised training samples x_i, y_i, where x_i is the input and y_i the acoustic label, our goal is to learn a multilingual model to estimate speaker labels for every language. Here, the subscript $i = 1, ..., N$ denotes the number of training samples, and superscript $l = 1, ..., L$ is the number of languages. Different languages share the same lower feature extraction layers, however, at higher levels, each language has their own SV layers.

We denote the feature extraction layers as G_f with parameters θ_f, and the SV layers as G_y with parameters θ_y, where l indexes the language. Since each language is considered a different domain, we attach Language Identification (LID) layers to the feature extraction layers to predict language labels. The LID layers are denoted as G_d with parameters θ_d. According to [6], the overall loss function for DAN is

$$L\left(\theta_f, \left\{\theta_y^l\right\}_{\ell=1,\ldots,L}, \theta_d\right) = (1-\lambda)\mathcal{L}_{SV}\left(\theta_f, \left\{\theta_y^l\right\}_{\ell=1,\ldots,L}\right) + \lambda L_{LID}\left(\theta_f, \theta_d\right), \tag{5}$$

where $\mathcal{L}_{LID}()$ is the average cross-entropy (CE) loss from context-dependent state classification for all languages, and $\mathcal{L}_{SV}()$ the loss of speaker verification, and $\lambda > 0$ is the adversarial weight which will be multiplied by the reversed gradients when it is backpropagated from the LID layer. Note that, when minimizing $\mathcal{L}_{SV}$, θ_y is always adjusted to minimize the speaker verification loss, and the error signals are backpropagated to optimize θ_f. On the other hand, G_f and G_d are jointly trained with an adversarial loss $\mathcal{L}_{LID}$, where θ_f is adjusted to maximize the loss, and θ_d is adjusted to minimize the loss. The two play a minimax game where G_d tries to discriminate inputs from different languages using features generated by G_f, while G_f tries to generate features to confuse G_d to not make the right domain classification decision (thanks to reverse gradients). The gradient reversal follows the implementation in [6], keeps the propagation unchanged in the forward path, and multiplies the gradient by $-\lambda$ during backpropagation. We update the parameters by back propagation using Adam

$$\begin{aligned} \theta_f \leftarrow \theta_f - \alpha \left(\frac{\partial \mathcal{L}_{SV}}{\partial \theta_f} - \lambda \frac{\partial \mathcal{L}_{LID}}{\partial \theta_f} \right), \\ \theta_y^l \leftarrow \theta_y^l - \alpha \frac{\partial \mathcal{L}_{SV}}{\partial \theta_y^l}, \\ \theta_d \leftarrow \theta_d - \alpha \lambda \frac{\partial \mathcal{L}_{LID}}{\partial \theta_d}. \end{aligned} \tag{6}$$

In which α is the learning rate.

The model architecture is illustrated in Figure 3.

3.6. Score adjustment techniques

3.6.1. Score normalization

Score normalization [18] is vital in producing well-calibrated decision scores for speaker verification tasks. Without the normalization, different distributions of target and non-target scores can be obtained for two different enrolled speaker models. For the same speaker model, the score distributions can vary due to differences in the testing (recording channel, acoustic conditions, language of the utterance, etc.) which calls for a condition-dependent threshold. In this work, we use as-norm [18], one of the most common score normalization techniques for the task of speaker verification.

In adaptive score normalization (as-norm), only part of the cohort is selected to compute mean and variance for normalization. We use the k highest scores in the cohort scores to compute mean and variance for and apply that result to compute z-norm and t norm.

3.6.2. Score calibration

Calibration [19] is the process of transforming probability scores emitted by a model so that their distribution match is observed in the training set. While the scores output by SV systems contain valuable information to separate the same speaker from the different-speaker trials, they cannot be interpreted in absolute terms, only relative to their distribution. Therefore, calibration is a necessary process to convert the output scores into useful absolute measures that can be interpreted and reliably thresholded to make decisions.

We implement a simple logistic regression model to calibrate the outputs, where the weights are optimized by the LBFGS [20] algorithm with a learning rate of 0.01, based on the labels of the public test from I-MSV Challenge 2022 Organizer.

3.6.3. Score fusion

State-of-the-art speaker verification systems take advantage of various base classifiers by fusing them to achieve reliable verification decisions. Fusion could be realized at the sensor level, feature level, or score level. Scores fusion is a favorable method due to its simplicity and good performance [21]. In our SV system, fusion [21, 22] is implemented as a linear combination of the base classifier scores.

All the scores S_i of the N subsystems are summed and averaged out to achieve the final score S_{final}

$$S_{final} = \frac{1}{N}\sum_{i=1}^{n} S_i. \tag{7}$$

Table 1: Public test equal error rate (EER) with the proposed models

Model	Back-end	Loss Function	Public test EER (%)
ECAPA	Cosine	AAMSoftmax	3.831
	PLDA	AAMSoftmax	3.684
	PLDA	LMCL	1.127
Adversarial ECAPA	PLDA	AAMSoftmax	2.263
	PLDA	LMCL	1.400
RawNet3	Cosine	AAMSoftmax	3.707
	PLDA	AAMSoftmax	2.913
Adversarial RawNet3	PLDA	AAMSoftmax	1.913
	PLDA	LMCL	1.463

Table 2: Private test EER results for the proposed systems

Model	Back-end	Loss function	Score normalization	Public test EER (%)	Private test EER (%)
ECAPA (1)	**PLDA**	**LMCL**	**No**	**1.127**	**2.973**
Adversarial ECAPA (2)	PLDA	LMCL	as-norm	1.406	-
Adversarial RawNet (3)	PLDA	LMCL	No	1.463	-
Ensemble 1+2+3	-	-	-	1.186	3.510
Ensemble 1+3	-	-	-	0.550	3.017

3.7. Scores post-processing

Based on the mechanism of EER evaluation, we propose a technique to post-process the calibrated scores of the systems. Specifically, the calibrated output is in the range of 0 to 1. We can further discriminate the predictions by pushing the scores to the absolute ends (0 and 1) by analysing the distribution of the scores for each test utterance.

For each test utterance u_i, we denote E_i as the set of enrollment utterances for that test audio. Each set E_i contains the enrollment audios E_{ij} where $j = 1..5$ in our problem. Say the similarity score between u_i and E_{ij} is $\text{Score}(u_i, E_{ij})$, if the highest similarity score (for one test utterance) is distinguished enough from the second-highest score (which is defined by a threshold η), we can add a small constant to the highest similarity score so that it can approach nearer to an absolute 1.0, which may yield better EER if that score falls in the true positive case. In contrast, the other scores for that test utterance will be reduced by a constant to approach nearer to an absolute 0.0.

That is, for each test utterance u_i with enrollment set E_i, if

$$\text{maxScore}(u_i, E_{ij}) - \text{secondMaxScore}(u_i, E_{ij}) > \eta,$$

$$\text{Score}(u_i, E_{ij})) = \begin{cases} \text{Score}(u_i, E_{ij})) + \epsilon & \text{if max,} \\ \text{Score}(u_i, E_{ij})) - \epsilon & \text{otherwise.} \end{cases} \tag{8}$$

In our model, we tuned η to be $5e - 4$.

4. EXPERIMENTAL SETUPS

4.1. ECAPA-TDNN

We used ECAPA-TDNN architecture with 512 channels in the convolutional frame layers. The SE-Block and attention module's bottleneck dimension is set to 128. The Res2Block's scale dimensions are set at 8. The final fully connected layer contains 192 nodes [3].

All models are trained using the triangular2 policy from [3] along with the Adam optimizer, with cyclical learning rates ranging between $1e - 8$ and $1e - 3$. 130k iterations make up one cycle's duration. The model was trained for 200 epochs. With a margin of 0.2 and a softmax prescaling of 30 for 4 cycles, all systems are trained using AAM-softmax. All of the weights in the model of $2e - 5$, except for the AAM-softmax weights, which utilize $2e - 4$, were decayed to avoid overfitting. For training, a mini-batch size of 128 is used [3].

4.2. RawNet3

For RawNet3's architecture, the parameterized filterbank layer has a kernel length of 251, stride size of 48, and 256 filters. 1024 filters are included in AFMS-Res2MP blocks, along with additional hyper-parameters including kernel length, pool size, and dilation. In addition to having 1024 filters, AFMS-Res2MP blocks also feature hyper-parameters for kernel length, pool size, and dilation. The Adam optimizer with scheduling for SGDR learning rates is used. The model is subjected to weight decay regularization of $5e - 5$ [23].

Every eight epochs, the learning rate resets and is set between $1e - 3$ and $5e - 6$. The model is trained for 200 epochs. Scale s is 30 and margin m is 0.3 for AAM-softmax. The model is trained using 3-second utterances that have been randomly clipped. A mini-batch is 512 in size [23].

4.3. Evaluation protocol

Extracting speaker embeddings and determining their score similarity (cosine or PLDA) are parts of the evaluation's routine procedure. For each speaker verification model, we provide the EER where the false acceptance rate equals the false rejection rate.

5. RESULTS

The performances on the public test of the two models EPACA-TDNN and RawNet3 under different back-end scores and different loss functions, without calibration applied on the outputs, are given in Table 1.

Based on the results table, the models using PLDA as a back-end score give much better results than cosine similarity. Moreover, the results are significantly improved when using the loss function LMCL. Therefore, the adversarial training focuses on the PLDA back-end and LMCL loss function to experiment for results. The findings indicate that, among the models, ECAPA with PLDA and LMCL provides the greatest outcome, whereas RawNet with any back-end score and loss function does not perform as well as the others. Therefore, with up to 5 submissions for private tests, we use 3 models: ECAPA, Adversarial ECAPA, Adversarial RawNet, and their ensemble.

The best models selected for private testing are shown in Table 2 with calibrated and post-processed outputs. The best and most stable result comes from the ECAPA model with PLDA, LMCL, and no score normalization achieves 1.1266% on the public test and 2.9734% on the private test. The ensemble results are quite good, but the performance is not as good as when calculating the public test set. The inferior results of the ensemble systems may lay in the fact that the utterances in the private test are far different from those of the public tests, and overfitting may have happened in the case of adversarial training, thus yielding worse results when ensembled with the ECAPA model.

6. CONCLUSIONS

In this paper, we presented our solution to the Constrained task of the COCOSDA INDIC-Multilingual Speaker Verification (I-MSV) Challenge 2022. The overall process comprises four stages: speaker embedding extraction, score normalization, calibration, and fusing score from base systems.

We have proposed an ensemble system based on RawNet3 and ECAPA. Experimental results show that the PLDA back-end model and Large margin cosine loss outperform other techniques of the same kind in identifying multilingual speakers. Moreover, adversarial training has also contributed to diminishing the language features of speakers, which resulted in improvements in the overall results when it comes to our multilingual problem.

Possible improvements in the future are expected to involve more sophisticated data preprocessing and extension of the adversarial layers to further discriminate the speakers better when more than one language is taken into account.

REFERENCES

[1] R. Auckenthaler, M. J. Carey, and J. S. Mason, “Language dependency in text-independent speaker verification,” in *2001 IEEE International Conference on Acoustics, Speech, and Signal-Processing. Proceedings (Cat. No. 01CH37221)*, vol. 1. IEEE, 2001, pp. 441–444.

[2] V. Peddinti, D. Povey, and S. Khudanpur, “A time delay neural network architecture for efficient modeling of long temporal contexts” in *Interspeech*, 2015, pp. 3214–3218.298.

[3] B. Desplanques, J. Thienpondt, and K. Demuynck, “ECAPA-TDNN: Emphasized channel attention, propagation and aggregation in tdnn based speaker verification,” *arXiv preprint arXiv:2005.07143*, 2020.

[4] Y. Jiang, K. A. Lee, Z. Tang, B. Ma, A. Larcher, and H. Li, “PLDA modeling in i-vector and supervector space for speaker verification,” in *Annual Conference of the International Speech-Communication Association (Interspeech)*, 2012.

[5] J. w. Jung, H. S. Heo, J. h. Kim, H. j. Shim, and H. J. Yu, "Rawnet: Advanced end-to-end deep neural network using raw waveforms for text-independent speaker verification," *arXiv preprintarXiv:1904.08104*, 2019.

[6] H. Wang, Y. Wang, Z. Zhou, X. Ji, D. Gong, J. Zhou, Z. Li, and W. Liu, "Cosface: Large margin cosine loss for deep face recognition," in *Proceedings of The IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5265–5274.

[7] M. Sato, H. Manabe, H. Noji, and Y. Matsumoto, "Adversarial training for cross-domain universal dependency parsing," in *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsingfrom Raw Text to Universal Dependencies*, 2017, pp. 71–79.

[8] B. J. Borgstr¨om, "Discriminative training of PLDA for speaker verification with x-vectors," *Department of Defense Under Air Force. URL:https://www.ll.mit.edu/sites/default/files/publication/doc/discriminative-PLDA-speaker-verification-borgstrom-121037. pdf [accessed 2024-02-27]*, 2020.

[9] G. Heigold, V. Vanhoucke, A. Senior, P. Nguyen, M. Ranzato, M. Devin, and J. Dean, "Multilingual acoustic models using distributed deep neural networks," in *2013 IEEE International-Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2013, pp. 8619–8623.

[10] J. T. Huang, J. Li, D. Yu, L. Deng, and Y. Gong, "Cross-language knowledge transfer using multilingual deep neural network with shared hidden layers," in *2013 IEEE International Conferenceon Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2013, pp. 7304–7308.

[11] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, and Laviolette, "Domainadversarial training of neural networks," *Journal of Machine Learning Research*, vol. 17, no. 59,pp. 1–35, 2016.

[12] K. Hu, H. Sak, and H. Liao, "Adversarial training for multilingual acoustic modeling," *arXiv preprint arXiv:1906.07093*, 2019.

[13] L. Li, R. Nai, and D. Wang, "Real additive margin softmax for speaker verification," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 7527–7531.

[14] Y. Liu, L. He, and J. Liu, "Large margin softmax loss for speaker verification," *arXiv preprint arXiv:1904.03479*, 2019.

[15] M. Ravanelli, T. Parcollet, P. Plantinga, A. Rouhe, S. Cornell, L. Lugosch, C. Subakan, N. Dawalatabad, A. Heba, J. Zhong et al., "Speechbrain: A general-purpose speech toolkit," *arXiv preprint arXiv:2106.04624*, 2021.

[16] A. Kanagasundaram, R. Vogt, D. Dean, S. Sridharan, and M. Mason, "I-vector based speaker recognition on short utterances" *International Speech Communication Association (ISCA)*, pp.2341–2344, 2011.

[17] 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.

[18] R. Auckenthaler, M. Carey, and H. Lloyd-Thomas, "Score normalization for text-independent speaker verification systems," *Digital Signal Processing*, vol. 10, no. 1-3, pp. 42–54, 2000.

[19] A. Shulipa, S. Novoselov, and Y. Matveev, "Scores calibration in speaker recognition systems," in *Speech and Computer: 18th International Conference, SPECOM 2016, Budapest, Hungary, August 23-27, 2016, Proceedings 18.* Springer, 2016, pp. 596–603.

[20] P. Moritz, R. Nishihara, and M. Jordan, "A linearly-convergent stochastic L-BFGS algorithm," in *Artificial Intelligence and Statistics. PMLR*, 2016, pp. 249–258.

[21] F. R˘astoceanu and M. Laz˘ar, "Score fusion methods for text-independent speaker verification applications," in *2011 6th Conference on Speech Technology and Human-Computer Dialogue (SpeD).* IEEE, 2011, pp. 1–6.

[22] V. Hautam¨aki, T. Kinnunen, F. Sedl´ak, K. A. Lee, B. Ma, and H. Li, "Sparse classifier fusion for speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21,no. 8, pp. 1622–1631, 2013.

[23] J. W. Jung, Y. J. Kim, H. S. Heo, B. J. Lee, Y. Kwon, and J. S. Chung, "Pushing the limits of raw waveform speaker recognition," *arXiv preprint arXiv:2203.08488*, 2022.

Received on May 10, 2023
Accepted on December 13, 2023