ATTENTION BACK-END FOR AUTOMATIC SPEAKER VERIFICATION WITH MULTIPLE ENROLLMENT UTTERANCES

Chang Zeng[1,2], Xin Wang[1], Erica Cooper[1], Xiaoxiao Miao[1], Junichi Yamagishi[1,2]

[1]National Institute of Informatics, Japan [2]SOKENDAI, Japan

ABSTRACT

Probabilistic linear discriminant analysis (PLDA) or cosine similarity have been widely used in traditional speaker verification systems as back-end techniques to measure pairwise similarities. To make better use of multiple enrollment utterances, we propose a novel attention back-end model that can be used for both text-independent (TI) and text-dependent (TD) speaker verification, and we use scaled-dot self-attention and feed-forward self-attention networks as architectures that learn the intra-relationships of enrollment utterances. To verify the proposed model, we conduct a series of experiments on the CNCeleb and VoxCeleb datasets by combining it with several state-of-the-art speaker encoders including TDNN and ResNet. Experimental results obtained using multiple enrollment utterances on CNCeleb show that the proposed attention back-end model leads to lower EER and minDCF scores than its PLDA and cosine similarity counterparts for each speaker encoder, and an experiment on VoxCeleb demonstrates that our model can be used even for a single enrollment case.

Index Terms— Speaker verification, TDNN, ResNet, Attention

1. INTRODUCTION

Automatic speaker verification (ASV) is a task that determines whether input speech is uttered by a claimed speaker or not. Until recently, generative models such as GMM-UBM [1] and i-vector [2] with Probabilistic Linear Discriminant Analysis (PLDA) [3] were dominant for this task. With the development of deep learning, these models have been replaced by neural networks. A good example is neural speaker embeddings [4, 5] extracted from a hidden layer of a neural speaker encoder network, such as x-vectors [6, 7], and it is reported that these have outperformed conventional i-vectors. For the case of x-vectors, speaker embedding vectors are extracted from the first fully-connected layer after a statistical pooling layer of TDNN [5]. PLDA is adopted as a back-end model for handling the channel mismatch between enrollment speakers and evaluation speech. In addition to TDNN, ResNet [8] with metric learning loss [9, 10] or discriminative training loss of softmax [11, 12] is also a popular choice for extracting speaker embeddings. Some of them use simple cosine similarity between enrollment speakers and evaluation speech as a back-end model.

The objective of this paper is to explore a new way of better handling multiple enrollment utterances and then propose a new back-end model that can use a varying number of embedding vectors corresponding to multiple enrollment utterances directly. In a real-world setting, enrollment speakers may provide multiple utterances to register their identity. We hypothesize that intra-variations of enrollment utterances contain important information such as expected acoustic variations of the enrolled speakers. Even though the traditional generative PLDA can combine multiple enrollment utterances

for a better representation of a speaker, there is still room for fully using intra-variations of enrollment utterances. Therefore, we propose a new back-end model based on scaled-dot self-attention and feed-forward self-attention networks to learn intra-variations.

In the proposed back-end model, multiple speaker embedding vectors are first stacked as a matrix that is used as the input for multi-head scaled-dot self-attention, which estimates the self-attention probabilities of the input matrix. The self-attention matrix is then multiplied with the speaker embedding matrix in order to get bias parameters that emphasize or de-emphasize not only multiple enrollment utterances but also individual dimensions of their embedding vectors. The second step is multi-head feed-forward self-attention for aggregating the matrix modified using the multi-head scaled-dot self-attention described above into one representative vector. Finally, cosine scoring using the representative vector and one extracted from an evaluation trial is done with a learnable logistic transform for better score calibration. This network is trained using the weighted loss of normal binary cross-entropy and a recently proposed generalized end-to-end (GE2E) loss for ASV [13]. Note that our attention back-end model, which uses attention mechanism on the *utterance-level*, is different from previous work [14, 15] in which attention mechanism was applied to *frame-level* aggregation.

Experiments are conducted on two ASV datasets, including the well explored VoxCeleb [16, 17] and new multi-genre CNCeleb [18, 19]. Our proposed attention-based back-end is first evaluated for a single enrollment case on VoxCeleb and further analyzed for a multiple enrollment case on CNCeleb. The remainder of the paper is organized as follows. Cosine similarity and PLDA are briefly reviewed in Section 2. Section 3 gives details on the proposed attention back-end model. The experimental setup and results are outlined in Section 4. Finally, conclusions are presented in Section 5.

2. CONVENTIONAL BACK-END MODELS

2.1. Cosine similarity

Cosine similarity can be used to evaluate how close of two speaker embeddings distributed in hyper-sphere and this serves as the simplest back-end. It has been frequently used as a scoring method for ASV systems trained using discriminative loss functions [9, 11, 20] that minimizes intra-class distance and maximizes inter-class distance. This scoring method is just an inter-product of a pair of speaker embedding vectors[1] to be scored and computed as follows.

$$P(\boldsymbol{e}_i, \boldsymbol{e}_j) = \frac{\boldsymbol{e}_i^\top \boldsymbol{e}_j}{||\boldsymbol{e}_i||\,||\boldsymbol{e}_j||} \tag{1}$$

As for the case of multi-session enrollment, which means using multiple enrollment utterances per speaker, a frequently made choice is to average speaker embedding $\boldsymbol{c}_n = \frac{1}{K}\sum_{k=1}^{K}\boldsymbol{e}_m$. Here, K is

---

[1]In this paper, all vectors are assumed to be column vectors.

the number of enrollment utterances per speaker. Alternatively, a speaker embedding may be extracted from concatenated waveform or acoustic features.

2.2. Probabilistic linear discriminant analysis

PLDA is also a frequently used back-end model. PLDA was first introduced to ASV in [21] to decompose the speaker embedding of an utterance n into a global mean $\boldsymbol{\mu}$, latent variable $\boldsymbol{h}_n$ and residual term $\boldsymbol{\epsilon}_n$ as shown below.

$$\boldsymbol{e}_n = \boldsymbol{\mu} + \boldsymbol{F}\boldsymbol{h}_n + \boldsymbol{\epsilon}_n \qquad (2)$$

In Gaussian PLDA (GPLDA), where both the latent variable $\boldsymbol{h}$ and residual term $\boldsymbol{\epsilon}$ are assumed to follow Gaussian distributions, that is, $\boldsymbol{h} \sim \mathcal{N}(0, \mathbf{I})$ and $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \boldsymbol{\Sigma})$, where $\mathbf{I}$ and $\boldsymbol{\Sigma}$ represent identity and covariance matrices, respectively, the score of a pair of speaker embeddings is given as follows for a single-session enrollment case:

$$P(\boldsymbol{e}_i, \boldsymbol{e}_j) = \boldsymbol{e}_i^\top \boldsymbol{Q} \boldsymbol{e}_i + \boldsymbol{e}_j^\top \boldsymbol{Q} \boldsymbol{e}_j + 2\boldsymbol{e}_i^\top \boldsymbol{P} \boldsymbol{e}_j \qquad (3)$$

$$\boldsymbol{P} = \boldsymbol{\Sigma}_{tot}^{-1} \boldsymbol{\Sigma}_{ac} (\boldsymbol{\Sigma}_{tot} - \boldsymbol{\Sigma}_{ac} \boldsymbol{\Sigma}_{tot}^{-1} \boldsymbol{\Sigma}_{ac})^{-1} \qquad (4)$$

$$\boldsymbol{Q} = \boldsymbol{\Sigma}_{tot}^{-1} - (\boldsymbol{\Sigma}_{tot} - \boldsymbol{\Sigma}_{ac} \boldsymbol{\Sigma}_{tot}^{-1} \boldsymbol{\Sigma}_{ac})^{-1} \qquad (5)$$

$$\boldsymbol{\Sigma}_{tot} = \boldsymbol{F}\boldsymbol{F}^\top + \boldsymbol{\Sigma} \qquad (6)$$

$$\boldsymbol{\Sigma}_{ac} = \boldsymbol{F}\boldsymbol{F}^\top, \qquad (7)$$

where $\{\boldsymbol{\mu}, \boldsymbol{F}, \boldsymbol{\Sigma}\}$ are trainable parameters and estimated on the basis of the EM algorithm.

There are several ways of handling the multi-session enrollment case with PLDA [22, 23]. According to results reported in [22, 23], we adopt the averaging and concatenating of multiple speaker embeddings as well as an extended PLDA formulation [23] as strategies to handle multiple enrollment utterances in our baseline model.

3. ATTENTION BACK-END

In this section, we explain how the proposed model uses attention mechanisms to measure pairwise similarity, including the detailed architecture of the model and its loss functions, which are illustrated in Fig. 1.

3.1. Model architecture

Suppose a speaker has K enrollment utterances. With a speaker encoder $\text{Enc}(\cdot)$, we get K enrollment embedding vectors $\{\boldsymbol{e}_1, \cdots, \boldsymbol{e}_K\}$, where $\boldsymbol{e}_k = \text{Enc}(\boldsymbol{x}_{1:T}^{(k)}) \in \mathbb{R}^D$, and $\boldsymbol{x}_{1:T}^{(k)}$ is the waveform data of the k-th enrollment utterance. Since K varies across speakers, we need to aggregate $\{\boldsymbol{e}_1, \cdots, \boldsymbol{e}_K\}$ into a single vector $\boldsymbol{h} \in \mathbb{R}^D$ using two different attention mechanisms in sequence. Specifically, after stacking a matrix $\boldsymbol{E} = [\boldsymbol{e}_1, \cdots, \boldsymbol{e}_K]^\top \in \mathbb{R}^{K \times D}$, we use multi-head scaled dot self-attention (SDSA) [24] to transform $\boldsymbol{E}$ into a hidden matrix $\boldsymbol{H} \in \mathbb{R}^{K \times D}$. With d_1 attention heads, we get

$$\boldsymbol{H} = \text{Concat}(\boldsymbol{H}_1, \boldsymbol{H}_2, \cdots, \boldsymbol{H}_{d_1})\boldsymbol{W}^O + \boldsymbol{E}, \qquad (8)$$

where the i-th head matrix $\boldsymbol{H}_i \in \mathbb{R}^{K \times \frac{D}{d_1}}$ is computed by

$$\boldsymbol{H}_i = \text{Attn}_1(\boldsymbol{Q}_i, \boldsymbol{K}_i, \boldsymbol{V}_i) = \text{Softmax}(\frac{\boldsymbol{Q}_i \boldsymbol{K}_i^\top}{\sqrt{D/d_1}})\boldsymbol{V}_i. \qquad (9)$$

The query, key, and value matrices are set to $\boldsymbol{Q}_i = \boldsymbol{E}\boldsymbol{W}_i^Q$, $\boldsymbol{K}_i = \boldsymbol{E}\boldsymbol{W}_i^K$, and $\boldsymbol{V}_i = \boldsymbol{E}\boldsymbol{W}_i^V$, respectively, and they all have the shape of $K \times \frac{D}{d_1}$. The matrices $\{\boldsymbol{W}_i^Q, \boldsymbol{W}_i^K, \boldsymbol{W}_i^V, \boldsymbol{W}^O\}$ are trainable parameters of the attention mechanism. The multiple heads in Eq.
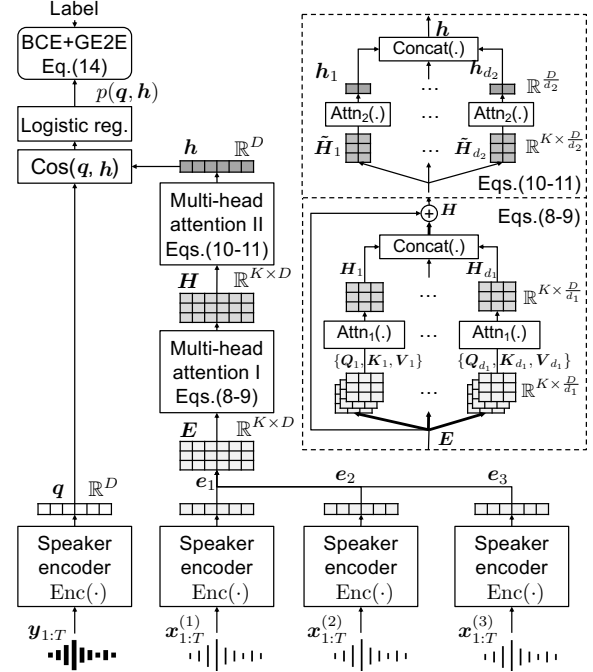

Fig. 1. Back-end architecture. Dashed boxes show detailed implementation of two multi-head attention blocks.

(8) are expected to attend to information from different representation subspaces at different positions.

Given $\boldsymbol{H}$ from Eq. (8), the second multi-head feed-forward self-attention (FFSA) mechanism aggregates it into the single representative embedding vector $\boldsymbol{h}$ by

$$\boldsymbol{h} = \text{Concat}(\boldsymbol{h}_1, \boldsymbol{h}_1, \cdots, \boldsymbol{h}_{d_2}). \qquad (10)$$

The j-th head vector $\boldsymbol{h}_j \in \mathbb{R}^{\frac{D}{d_2}}$ is computed as

$$\boldsymbol{h}_j = \text{Attn}_2(\tilde{\boldsymbol{H}}_j) = \text{Softmax}(\boldsymbol{v}_j^\top \text{Tanh}(\boldsymbol{W}_j \tilde{\boldsymbol{H}}_j^\top))\tilde{\boldsymbol{H}}_j, \qquad (11)$$

where $\boldsymbol{v}_j \in \mathbb{R}^{D_2}$ and $\boldsymbol{W}_j \in \mathbb{R}^{D_2 \times \frac{D}{d_2}}$ are trainable parameters, and $\tilde{\boldsymbol{H}}_j \in \mathbb{R}^{K \times \frac{D}{d_2}}$ is a sub-matrix of $\boldsymbol{H}$, i.e., $\boldsymbol{H} = [\tilde{\boldsymbol{H}}_1, \tilde{\boldsymbol{H}}_2, \cdots, \tilde{\boldsymbol{H}}_{d_2}]$. This is inspired by the multi-head attention-based aggregation proposed in [25].

After obtaining the representative vector for an enrollment speaker, the cosine similarity score between an evaluation speaker embedding, which is denoted as $\boldsymbol{q} = \text{Enc}(\boldsymbol{y}_{1:T})$, and the representative vector $\boldsymbol{h}$ is calculated. Further, a logistic regression (LR) is used for better calibration as shown below:

$$P(\boldsymbol{q}, \boldsymbol{h}) = \frac{1}{1 + \exp^{-s}} = \frac{1}{1 + \exp^{-a\text{Cos}(\boldsymbol{q}, \boldsymbol{h}) - b}} \qquad (12)$$

$$\text{Cos}(\boldsymbol{q}, \boldsymbol{h}) = \frac{\boldsymbol{q}^\top \boldsymbol{h}}{||\boldsymbol{q}|| \, ||\boldsymbol{h}||}, \qquad (13)$$

where $P(\boldsymbol{q}, \boldsymbol{h})$ denotes the probability of $\boldsymbol{q}$ and $\boldsymbol{h}$ belonging to the same speaker and also serves as the final score for decision making, and a and b are trainable parameters.

Table 1. Composition of pairs of (test-trial, enrollment-data) for training back-end model and ground-truth labels from mini-batch. A, B, and C are speaker IDs, and 1, 2, 3 and 4 are his or her audio IDs. ✓and ×denote test and enrollment audio files, respectively.

		A				B				C			Test	Enroll	Label
	1	2	3	4	1	2	3	4	1	2	3	4			
	✓	×	×	×									q_{A1}	h_{A1}	P
	✓					×	×	×					q_{A1}	h_{B1}	N
	✓								×	×	×		q_{A1}	h_{C1}	N
	×	✓	×	×									q_{A2}	h_{A2}	P
		✓			×		×	×					q_{A2}	h_{B2}	N
		✓							×		×	×	q_{A2}	h_{C2}	N
						⋮									
	×	×	×									✓	q_{C4}	h_{A4}	N
					×	×	×					✓	q_{C4}	h_{B4}	N
									×	×	×	✓	q_{C4}	h_{C4}	P

(left vertical label: **Trials to be used for training**)

3.2. Loss functions

In the training stage, speakers in a mini-batch can be seen as either target or non-target speakers, and all audio files included in one mini-batch can be used to define various positive and negative pairs. Table 1 illustrates one example of forming pairs of *(test-trial, enrollment-data)*. In this example, one mini-batch of data has 3 speakers A, B, and C, and each speaker has 4 embeddings whose index ranges from 1 to 4. There are numerous ways of composing the pairs, but we here consider only the following cases. For positive pairs where the test trial and enrollment data are from the same speaker, one test trial is selected from the speaker's data, and the rest are left for enrollment. For negative pairs where the speaker of the test trial is different from that of the enrollment data, we consider only pairs marked by (test-trial=✓, enroll=(×, ×, ×)) of other speakers included in a mini-batch.

Using the above positive and negative pairs included in a mini-batch, we optimize the proposed attention back-end. The loss function for the optimization is a weighted sum of binary cross-entropy (BCE) and generalized end-to-end (GE2E) loss [13] functions given below to train our model.

$$\mathcal{L} = \lambda \mathcal{L}_{\text{ge2e}} + (1 - \lambda)\mathcal{L}_{\text{bce}} \qquad (14)$$

As shown in Table 1, let us define q_{lm} as a speaker embedding vector extracted from the m-th test trial of speaker l and h_{nm} as a vector h of Eq. (10) based on an enrollment set m, which contains multiple audio files uttered by speaker n. The BCE loss can be computed as

$$\mathcal{L}_{\text{bce}} = - \sum_{\forall l,m,n} [\mathcal{I}(l = n) \log P(q_{lm}, h_{nm}) \qquad (15)$$
$$+ \mathcal{I}(l \neq n) \log (1 - P(q_{lm}, h_{nm}))],$$

where $\mathcal{I}(\cdot)$ is an indicator function that returns one when its argument is true and zero otherwise. In addition to BCE, GE2E [13] is also used for reducing intra-class variance and enlarging inter-class variance. In the original GE2E loss, an averaged speaker embedding is computed per speaker as a centroid. Here, we treated a vector h of Eq. (10) as a centroid instead of a simple mean. Given this revision, the GE2E loss is given by

$$\mathcal{L}_{\text{ge2e}} = - \sum_{\forall l,m} \log \frac{\exp^{P(q_{lm}, h_{lm})}}{\sum_{\forall n} \exp^{P(q_{lm}, h_{nm})}}. \qquad (16)$$

Table 2. VoxCeleb verification performance results with TDNN and different back-ends

Back-end	EER(%)	minDCF(0.01)	minDCF(0.001)
Cosine	10.51	0.7928	0.8718
PLDA	**3.14**	0.3456	0.5567
Proposed	3.26	**0.3323**	**0.5134**

4. EXPERIMENTS

Datasets We conducted our experiments on two totally different datasets: VoxCeleb1&2 and CNCeleb1&2, which are English and Chinese datasets, respectively. VoxCeleb1&2 contains 7,363 speakers, whose voice comes from interviews, and is split into 7,323 speakers for training data and 40 speakers for testing data according to the official trial protocol. For trial pairs in the VoxCeleb1 test data, each speaker only has one enrollment utterance. Different from VoxCeleb, the CNCeleb1&2 dataset includes in-the-wild speech utterances of 3,000 speakers in 11 different genres, which leads to complex and composite inter-session variations, both intrinsic (i.e., speaking style, physiological status) and extrinsic (i.e., recording device, background noise) [19]. The official protocol uses 2,800 speakers for training and 200 speakers whose identity was enrolled in multiple sessions for evaluation. MUSAN [26] and RIRs [27] datasets were both used to augment the above datasets.

Baseline Our experiments used a few well-testified DNN models to extract fixed-length speaker embeddings: TDNN [6], TDNN with attentive statistics pooling (TDNN-ASP) [14], ECAPA-TDNN [28], and ResNet [8]. All TDNN variants used plain softmax loss, while the ResNet used am-softmax loss [20]. The TDNN models were trained by using an SGD optimizer with 0.01 learning rate. As for ResNet, it was trained by using an SGD optimizer with 0.1 learning rate. Both of them were trained for 20 epochs with 0.1 learning rate decay every 8 epochs. During the back-end training, all the speaker encoders were frozen. The GPLDA back-end was trained by using a subset of training data, which consisted of 100,000 speaker embeddings selected randomly from the training data. First, a linear discriminative analysis (LDA) transformation matrix was trained to reduce the dimension of the speaker embedding from 512 to 400. Then, the output of LDA was used to train a GPLDA model in which the dimension of the latent factor is 150.

Back-end training A balanced batch strategy was adopted to train the attention back-end model. For each mini-batch, assuming it has M speakers and K speaker embeddings per speaker, the size of one mini-batch was $M \times K$. In our experiment, M and K were set to 256 and 5, respectively. When one mini-batch of data was fed into the back-end model, it was rearranged to form speaker verification trials that had multiple enrollment utterances (or embedding vectors) as described in Section 3.2 and the ratio of positive samples to negative samples was $1 : (M - 1)$. The proposed model was trained for 40 epochs by using an SGD optimizer with a cyclical learning rate scheduler [29], which changed learning rate from 0.00001 to 0.00003 every 2,000 updating steps. The hyper-parameter λ, which is the weight of two loss functions, was set to 0.6. In the testing phase, the output of Eq. (12) served as a decision score.

4.1. Results

Although the main advantage of the proposed method is that multiple enrollment utterances can be used, we first confirmed the performance of the proposed back-end using a single enrollment case on the VoxCeleb dataset. This ensures the generalization capability of our proposed back-end with respect to the number of enrollment utterances. Table 2 shows the result for the TDNN speaker encoder on

Table 3. CNCeleb verification performance results of two encoders with different back-ends

Spk encoder	Back-end	Enroll process.	EER(%)	minDCF(0.01)
TDNN	Cosine	Concat	20.82	0.8114
TDNN	Cosine	Mean	19.86	0.8350
TDNN [19]	PLDA	Concat	12.52	-
TDNN	PLDA	Concat	12.09	0.6105
TDNN	PLDA	Mean	13.29	0.6522
TDNN	PLDA	Multi-session	21.35	0.8417
TDNN	Proposed	Attention	**10.12**	**0.5649**
ResNet	Cosine	Concat	12.46	0.5484
ResNet	Cosine	Mean	11.86	0.5045
ResNet	PLDA	Concat	15.66	0.5890
ResNet	PLDA	Mean	14.56	0.5692
ResNet	PLDA	Multi-session	21.91	0.8082
ResNet	Proposed	Attention	**10.77**	**0.4983**
TDNN-ASP	PLDA	Concat	11.63	0.6157
TDNN-ASP	PLDA	Mean	10.67	0.5990
TDNN-ASP	Proposed	Attention	**9.90**	**0.5606**
ECAPA-TDNN	PLDA	Concat	10.54	0.5806
ECAPA-TDNN	PLDA	Mean	9.51	0.5665
ECAPA-TDNN	Proposed	Attention	**8.93**	**0.5043**

the VoxCeleb dataset. We can see that the attention back-end gave a number comparable to that of PLDA in terms of EER. Moreover, in terms of minDCF(0.01 and 0.001), the proposed model slightly outperformed PLDA model by 3.8% and 7.8%, respectively, in relative. Although the proposed architecture is motivated by the use of multiple enrollment utterances, it also works reasonably well for the single enrollment case.

In the following experiments, we focus on CNCeleb since this dataset has multiple enrollment utterances per speaker, and the acoustic conditions of the enrollment data may vary significantly. Therefore, it is a suitable dataset for evaluating the proposed methods. The first and the second groups of Table 3 shows the EER and minDCF(0.01) results for various back-end models using TDNN or ResNet encoders trained on CNCeleb, respectively. The third line of this table is the best result of others' work that we are aware of so far for this dataset. From the table, we first see that PLDA variants were better than cosine similarity variants for the TDNN encoder, whereas cosine similarity variants were more suitable for the discriminatively trained ResNet encoder, as reported in the literature. We can then see that our attention back-end model had a lower EER and minDCF score than the PLDA and cosine similarity variants for each speaker encoder type. The DET curves of these two speaker encoders with different back-end methods are shown in Figure 2.

We also combined our attention back-end with TDNN-ASP and ECAPA-TDNN models, which contained frame-level attention module. The result for the third group of Table 3 indicates that the utterance-level attention module included in our proposed model was complementary to the frame-level attention used in the speaker encoders in the third group. Also, we can confirm that our attention back-end had better performance than PLDA back-end again.

4.2. Ablation study

To explore the contribution of each component of our back-end for performance improvement, we conducted an ablation experiment whose result is shown in Table 4. This study also used the CNCeleb dataset. The proposed model can be divided into five components (SDSA, FFSA, LR, BCE and GE2E) as shown in the first line of the table. In each ablation study, we removed one component from the model except the FFSA, which was substituted with a simple mean

Fig. 2. DET curves for combinations of TDNN and ResNet speaker encoders and several different back-end models.
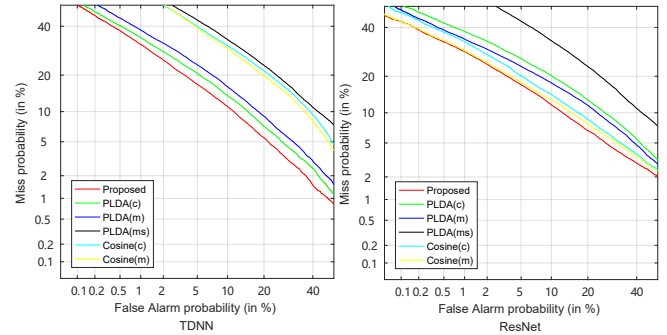

Table 4. Ablation experiment results on CNCeleb with TDNN speaker encoder.

Systems	EER(%)	DCF(0.01)	DCF(0.001)
SDSA + FFSA + LR + BCE + GE2E	10.12	0.5649	0.7127
FFSA + LR + BCE + GE2E	19.55	0.8095	0.9046
SDSA + mean + LR + BCE + GE2E	10.26	0.5876	0.7270
SDSA + FFSA + BCE + GE2E	48.32	1.0000	1.0000
SDSA + FFSA + LR + GE2E	12.74	0.5937	0.7350
SDSA + FFSA + LR + BCE	10.28	0.5896	0.7402

operation for aggregation.

From the second and the third lines of the table, it was confirmed that two self-attention mechanisms both contributed to the improvement. Concretely, both SDSA for summarizing intra-variations of multiple speaker embeddings and FFSA for aggregation contributed to the performance, and SDSA seemed to have a more important role. Additionally, from the fourth line, we can also see that LR for calibration was another essential component of the proposed model. The first, fifth and sixth lines of the table indicates interesting tendencies of our loss function using the weighted sum of BCE and GE2E. From a comparison of the fifth and sixth lines, we see that the proposed model using BCE loss only had a lower EER than that using GE2E loss only, while one using GE2E loss resulted in a lower minDCF(0.001) value. Combining BCE and GE2E thus resulted in the lowest values for all the metrics.

5. CONCLUSION

In this paper, we proposed a new attention back-end model for speaker verification using a varying number of multiple enrollment utterances and compared it with other back-end methods. The proposed back-end uses two types of attention, SDSA and FFSA, and they are trained based on weighted BCE and GE2E loss. For the CNCeleb dataset, the proposed attention back-end model was combined with several speaker encoders including TDNN and ResNet, and they were compared with PLDA or cosine similarity counterparts. The proposed model showed superiority and resulted in a lower EER and minDCF score than the PLDA and cosine similarity methods for each speaker encoder type. The experimental results on the VoxCeleb dataset show that the proposed back-end has comparable performance to conventional PLDA scoring even for the case of a single enrollment utterance.

6. REFERENCES

[1] Douglas A Reynolds, Thomas F Quatieri, and Robert B Dunn, "Speaker verification using adapted gaussian mixture models," *Digital signal processing*, vol. 10, no. 1-3, pp. 19–41, 2000.

[2] Najim Dehak, Patrick J Kenny, Réda Dehak, Pierre Dumouchel, and Pierre Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788–798, 2010.

[3] Sergey Ioffe, "Probabilistic linear discriminant analysis," in *European Conference on Computer Vision*. Springer, 2006, pp. 531–542.

[4] Ehsan Variani, Xin Lei, Erik McDermott, Ignacio Lopez Moreno, and Javier Gonzalez-Dominguez, "Deep neural networks for small footprint text-dependent speaker verification," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2014, pp. 4052–4056.

[5] David Snyder, Daniel Garcia-Romero, Daniel Povey, and Sanjeev Khudanpur, "Deep neural network embeddings for text-independent speaker verification.," in *Interspeech*, 2017.

[6] David Snyder, Daniel Garcia-Romero, Gregory Sell, Daniel Povey, and Sanjeev Khudanpur, "X-vectors: Robust dnn embeddings for speaker recognition," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5329–5333.

[7] Desh Raj, David Snyder, Daniel Povey, and Sanjeev Khudanpur, "Probing the information encoded in x-vectors," in *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2019, pp. 726–733.

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[9] Chao Li, Xiaokong Ma, Bing Jiang, Xiangang Li, Xuewei Zhang, Xiao Liu, Ying Cao, Ajay Kannan, and Zhenyao Zhu, "Deep speaker: an end-to-end neural speaker embedding system," *arXiv preprint arXiv:1705.02304*, vol. 650, 2017.

[10] Joon Son Chung, Jaesung Huh, Seongkyu Mun, Minjae Lee, Hee Soo Heo, Soyeon Choe, Chiheon Ham, Sunghwan Jung, Bong-Jin Lee, and Icksang Han, "In defence of metric learning for speaker recognition," in *Interspeech*, 2020.

[11] Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song, "Sphereface: Deep hypersphere embedding for face recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 212–220.

[12] Feng Wang, Jian Cheng, Weiyang Liu, and Haijun Liu, "Additive margin softmax for face verification," *IEEE Signal Processing Letters*, vol. 25, no. 7, pp. 926–930, 2018.

[13] Li Wan, Quan Wang, Alan Papir, and Ignacio Lopez Moreno, "Generalized end-to-end loss for speaker verification," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 4879–4883.

[14] Koji Okabe, Takafumi Koshinaka, and Koichi Shinoda, "Attentive statistics pooling for deep speaker embedding," in *Interspeech*, 2018.

[15] Yingke Zhu, Tom Ko, David Snyder, Brian Mak, and Daniel Povey, "Self-attentive speaker embeddings for text-independent speaker verification.," in *Interspeech*, 2018.

[16] A. Nagrani, J. S. Chung, and A. Zisserman, "Voxceleb: a large-scale speaker identification dataset," in *Interspeech*, 2017.

[17] J. S. Chung, A. Nagrani, and A. Zisserman, "Voxceleb2: Deep speaker recognition," in *Interspeech*, 2018.

[18] Yue Fan, JW Kang, LT Li, KC Li, HL Chen, ST Cheng, PY Zhang, ZY Zhou, YQ Cai, and Dong Wang, "Cn-celeb: a challenging chinese speaker recognition dataset," in *2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 7604–7608.

[19] Lantian Li, Ruiqi Liu, Jiawen Kang, Yue Fan, Hao Cui, Yunqi Cai, Ravichander Vipperla, Thomas Fang Zheng, and Dong Wang, "Cn-celeb: multi-genre speaker recognition," *arXiv preprint arXiv:2012.12468*, 2020.

[20] Hao Wang, Yitong Wang, Zheng Zhou, Xing Ji, Dihong Gong, Jingchao Zhou, Zhifeng Li, and Wei 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.

[21] Daniel Garcia-Romero and Carol Y Espy-Wilson, "Analysis of i-vector length normalization in speaker recognition systems," in *Interspeech*, 2011.

[22] Kong Aik Lee, Anthony Larcher, Chang Huai You, Bin Ma, and Haizhou Li, "Multi-session plda scoring of i-vector for partially open-set speaker detection," in *Interspeech*, 2013.

[23] Padmanabhan Rajan, Anton Afanasyev, Ville Hautamäki, and Tomi Kinnunen, "From single to multiple enrollment i-vectors: Practical plda scoring variants for speaker verification," *Digital Signal Processing*, vol. 31, pp. 93–101, 2014.

[24] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin, "Attention is all you need," in *NIPS*, 2017.

[25] Zhouhan Lin, Minwei Feng, Cicero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio, "A structured self-attentive sentence embedding," in *ICLR*, 2017.

[26] David Snyder, Guoguo Chen, and Daniel Povey, "MUSAN: A Music, Speech, and Noise Corpus," 2015, arXiv:1510.08484v1.

[27] Tom Ko, Vijayaditya Peddinti, Daniel Povey, Michael L Seltzer, and Sanjeev Khudanpur, "A study on data augmentation of reverberant speech for robust speech recognition," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 5220–5224.

[28] Brecht Desplanques, Jenthe Thienpondt, and Kris Demuynck, "Ecapa-tdnn: Emphasized channel attention, propagation and aggregation in tdnn based speaker verification," in *Interspeech*, 2020.

[29] Leslie N Smith, "Cyclical learning rates for training neural networks," in *2017 IEEE winter conference on applications of computer vision (WACV)*. IEEE, 2017, pp. 464–472.