Deep Segment Attentive Embedding for Duration Robust Speaker Verification

Bin Liu*†,Shuai Nie*, Wenju Liu*,Hui Zhang‡, Xiangang Li‡ and Changliang Li§
* National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, China
E-mail: {bin.liu2015,shuai.nie,lwj}@nlpr.ia.ac.cn
† School of Artificial Intelligence, University of Chinese Academy of Sciences, China
‡ DiDi AI Labs, China
E-mail: {ethanzhanghui,lixiangang}@didiglobal.com
§ kingsoft AI lab, China
E-mail: lichangliang@kingsoft.com

Abstract—**Deep learning based speaker verification usually uses a fixed-length local segment randomly truncated from an utterance to learn the utterance-level speaker embedding, while using the average embedding of all segments of a test utterance to verify the speaker, which results in a critical mismatch between testing and training. This mismatch degrades the performance of speaker verification, especially when the durations of training and testing utterances are very different. To alleviate this issue, we propose the deep segment attentive embedding method to learn the unified speaker embeddings for utterances of variable duration. Each utterance is segmented by a sliding window and LSTM is used to extract the embedding of each segment. Instead of only using one local segment, we use the whole utterance to learn the utterance-level embedding by applying an attentive pooling to the embeddings of all segments. Moreover, the similarity loss of segment-level embeddings is introduced to guide the segment attention to focus on the segments with more speaker discriminations, and jointly optimized with the utterance-level embeddings loss. Systematic experiments on DiDi Speaker Dataset, Tongdun and VoxCeleb show that the proposed method significantly improves system robustness and achieves the relative EER reduction of 18.3%, 50% and 11.54% , respectively.**

I. Introduction

The key to speaker verification is to extract the utterance-level speaker vectors with a fixed dimension for utterances of variable duration. The extracted speaker vector is expected to be as close as possible to the same speaker while far from other speakers. It remains a challenge to extract the robust speaker vectors for utterances of variable duration, especially when the utterance duration varies greatly. The i-vector/PLDA framework [1], [2], [3] can easily extract the fixed dimension speaker vectors for utterances of arbitrary duration using statistical modeling. But it suffers performance reduction when handling short utterances [4], [5]. The reason is that i-vector is a Gaussian-based statistical feature, whose estimation need sufficient samples. And the short utterance will lead to the uncertainty in the estimated i-vector.

Deep learning based speaker embedding [4], [6], [7] is another mainstream approach to speaker verification, which has been extensively studied recently and achieved promising performance in short-duration text-independent task. There are two ways to extract speaker embeddings using deep models. One approach is averaging bottleneck features from frame-level speaker classification networks [6]. Another approach is directly learning utterance-level speaker embeddings with distance-based similarity loss, such as triplet loss [4], [8] and generalized end-to-end (GE2E) loss [7].

LSTM-based speaker embedding is one of the most important deep speaker verification methods and has been demonstrated to be substantially promising [9], [10]. Owing to the powerful ability in modeling time-series data, LSTM can effectively capture the local correlation information of speech, which is very important for speaker verification. But it is still challenging for LSTM to model the long-term dependency of utterances, especially very long utterances. In addition, in order to facilitate batch training, LSTM-based speaker verification usually uses a fixed-length local segment randomly truncated from an utterance to learn the utterance-level speaker embedding in training phase, while using the average embedding of all segments of a test utterance to verify the speaker in testing phase, which leads to a critical mismatch between testing and training. The mismatch dramatically degrades the performance of speaker verification, especially when the difference of durations between training and testing utterances is large. Many methods are proposed to handle the issue of duration variability. The attention-based pooling [11], [12] is one of the most important technologies. But most of the attention mechanisms are performed at the frame level, which will lead to the "over-average" problem, especially when the utterance is very long.

To alleviate this issue, we propose the deep segment attentive embedding method to learn the unified speaker embeddings for utterances of variable duration. For both training and testing, we use a sliding window to divide utterances into the fixed-length segments and then use LSTM to extract the embedding of each segment. Finally, all segment-level embeddings of an utterance are pooled into a fixed-dimension vector through the segment attention, which is used as the utterance-level speaker embedding. The similarity loss of utterance-level embeddings is used to train the whole network. In addition, in order to guide the segment attention to focus on the segments with more speaker discriminations, we further incorporate the similarity loss o embeddings. With the joint optimization of th and utterance-level similarity loss, both local d information of utterances are taken into acc only using one local segment, we use the wh learn the utterance-level embedding, which un of training and testing and avoids the mismatc

II. Related Work

There are some efforts on the issue of du ity. For example, in the conventional i-vecto proposed to propagate the uncertainty relevan extraction process into the PLDA model, whic the duration variability. Moreover, in the deep speaker embedding systems, the complementa proposed in [14], [15], [16] in order to solve large variation in text-independent utterance duration variation. It acts as a regularizer t intra-class distance variance of the final emb However, they don't explicitly model the dur of utterances and the mismatch between train phase still exists.

Furthermore, attention mechanisms have been utilized to capture the long-term variations of speaker characteristics in [11], [12]. An important metric is computed by the attention network, which is used to calculate the weighted mean of the frame-level embedding vectors. However, most of the attention mechanisms are performed at the frame level, which will lead to the "over-average" problem, especially when the utterance is very long.

III. Proposed Approach

It is still challenging for LSTM to model the long-term dependency of utterances, especially very long utterances. And the mismatch between training and testing phase degrades the performance of speaker verification, especially when the difference of durations between training and testing utterances is large. Therefore, we propose the deep segment attentive embedding method to extract the unified speaker embeddings for utterances of variable duration.

As is shown in Fig. 1, we use a sliding window with 50% overlap to divide utterances into the fixed-length segments and LSTM is used to extract the embedding of each segment. Finally, all segment-level embeddings of an utterance are pooled into a fixed-dimension utterance-level speaker embedding through the segment attention mechanism. The whole network is trained with the joint supervision of the utterance-level and segment-level similarity loss.

A. Deep segment attentive embedding

For both training and testing, we use a sliding window with 50% overlap to divide an utterance into the fixed-length segments. Supposed that we get N speech segments $\mathbf{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_N\}$. The sliding window length T is randomly chosen within $[80, 120]$ frames but the length of segments in a batch is fixed. The vector $\mathbf{x}_n^t$ represents the

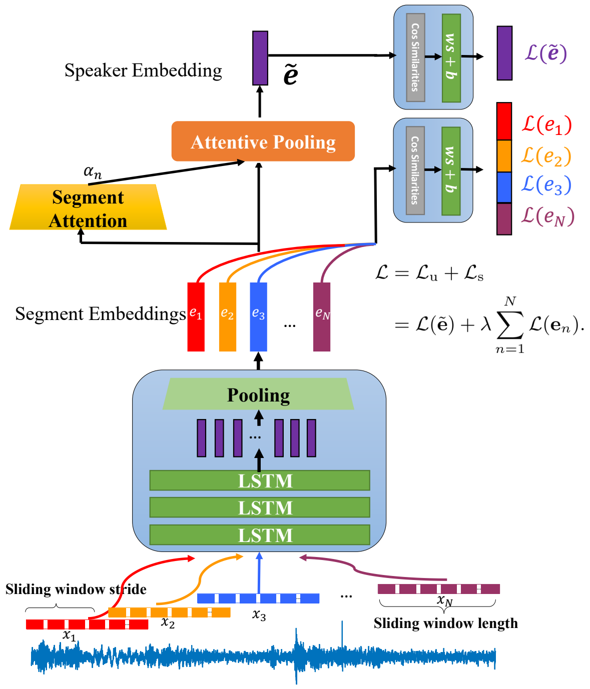

Fig. 1. System overview. For each training batch, there are $Q \times P$ utterances from Q different speakers and each speaker has P utterances. We only draw one utterance for simplicity.

feature of segment n at frame t, which is fed into the network and the output is $\mathbf{h}_n^t$. The last frame of output is used as the segment representation $f(\mathbf{x}_n; \mathbf{w}) = \mathbf{h}_n^T$, where $\mathbf{w}$ represents parameters of the network. The segment-level speaker embedding is defined as the L_2 normalization of the segment representation:

$$\mathbf{e}_n = \frac{f(\mathbf{x}_n; \mathbf{w})}{\|f(\mathbf{x}_n; \mathbf{w})\|_2}. \tag{1}$$

We compute the embedding vector of each segment $\mathbf{E} = [\boldsymbol{e}_1, \boldsymbol{e}_2, \cdots, \boldsymbol{e}_N]$ according to Eq. (1). Let the dimension of the segment-level speaker embedding $\boldsymbol{e}_n$ be d_e and $\mathbf{E} \in \mathbb{R}^{N \times d_e}$.

It is often the case that some segment-level embeddings are more relevant and important for discriminating speakers than others. We therefore apply attention mechanisms to integrate the segment embeddings by automatically calculating the importance of each segment. For each segment-level embedding $\boldsymbol{e}_n$, we could learn a score $\boldsymbol{\alpha}_n$ using the segment attention mechanism. All segment-level embeddings of an utterance are pooled into a fixed-dimension utterance-level speaker embedding through the segment attention mechanism.

For each segment embedding $\boldsymbol{e}_n$, we apply the multi-head attention mechanism [17] to learn a score $\boldsymbol{\alpha}_n$ as follows:

$$\boldsymbol{\alpha}_n = \mathrm{softmax}\left(g(\boldsymbol{e}_n \mathbf{W}_1)\mathbf{W}_2\right), \tag{2}$$

where $\mathbf{W}_1$ and $\mathbf{W}_2$ are parameters of the multi-head attention mechanism; $\mathbf{W}_1 \in \mathbb{R}^{d_e \times d_a}$; $\mathbf{W}_2 \in \mathbb{R}^{d_a \times d_r}$; d_a is the attention dim and d_r is a hyperparameter that represents the number of attention heads; $g(\cdot)$ is the ReLU activation function [18]. When the number of attention heads $d_r = 1$, it is simply a basic attention. The normalized weight $\boldsymbol{\alpha}_n \in [0,1]^{d_r}$ is computed by the softmax function. The weight vector is then used in the attentive pooling layer to calculate the utterance-level speaker embedding $\tilde{\boldsymbol{e}}$:

$$\tilde{\boldsymbol{e}} = \mathbf{A}^{\mathrm{T}}\mathbf{E}, \tag{3}$$

where $\mathbf{A} = [\boldsymbol{\alpha}_1, \cdots, \boldsymbol{\alpha}_N] \in \mathbb{R}^{N \times d_r}$ is the attention matrix and $\tilde{\boldsymbol{e}} \in \mathbb{R}^{d_r \times d_e}$. We concat d_r embeddings to build the the utterance-level speaker embedding.

When the number of attention heads $d_r = 1$, $\tilde{\boldsymbol{e}}$ is simply a weighted mean vector computed from $\mathbf{E}$, which is expected to reflect an aspect of speaker discriminations in the given utterance. Obviously, speakers can be discriminated along multiple aspects, especially when the utterance duration is long. By increasing d_r, we can easily have multiple attention heads to focus on different pattern aspects from an utterance. In order to encourage diversity in the attention vectors, [12] introduced a penalty term $\mathcal{L}_p$ when $d_r > 1$:

$$\mathcal{L}_p = \left\| \mathbf{A}^T\mathbf{A} - \mathbf{I}_n \right\|_F^2, \tag{4}$$

where $\mathbf{I}_n$ is the identity matrix where $n = d_r$ denotes the number of rows and columns and $\|\cdot\|_F$ represents the Frobenius norm of a matrix. $\mathcal{L}_p$ can encourage each attention head to extract different information from the same utterance. It is similar to L_2 regularization and is minimized together with the original cost of the system.

B. Loss function

After obtaining the utterance-level speaker embedding, we calculate the similarity loss using the generalized end-to-end (GE2E) loss formulation [7]. The GE2E loss is based on processing a large number of utterances at once to minimize the distance of the same speaker while maximizing the distance of different speakers.

For each batch training, we randomly choose $Q \times P$ utterances from Q different speakers with P utterances per speaker. And we calculate the utterance-level speaker embedding $\tilde{\boldsymbol{e}}_{ji}$ based on Eqs. (1) to (3) for each utterance. $\tilde{\boldsymbol{e}}_{ji}$ represents the speaker embedding of the j^{th} speaker's i^{th} utterance. And the centroid of embedding vectors from the j^{th} speaker is defined:

$$\mathbf{c}_j = \mathbf{E}_i\left[\tilde{\boldsymbol{e}}_{ji}\right] = \frac{1}{P}\sum_{i=1}^{P}\tilde{\boldsymbol{e}}_{ji}. \tag{5}$$

GE2E builds a similarity matrix $\mathbf{S}_{ji,k}$ that defines the scaled cosine similarities between each embedding vector $\tilde{\boldsymbol{e}}_{ji}$ to all centroids $\mathbf{c}_k$ ($1 \leqslant j, k \leqslant Q$ and $1 \leqslant i \leqslant P$):

$$\mathbf{S}_{ji,k} = w \cdot \cos(\tilde{\boldsymbol{e}}_{ji}, \mathbf{c}_k) + b, \tag{6}$$

where w and b are learnable parameters. The weight is constrained to be positive $w > 0$, because the scaled similarity is expected to be larger when the cosine similarity is larger.

During the training, each utterance's embedding is expected to be similar to the centroid of that utterance's speaker, while far from other speakers' centroids. The loss on each speaker embedding $\tilde{\boldsymbol{e}}_{ji}$ could be defined as:

$$\begin{aligned}\mathcal{L}(\tilde{\boldsymbol{e}}_{ji}) &= -\log\frac{\exp(\mathbf{S}_{ji,j})}{\sum_{k=1}^{Q}\exp(\mathbf{S}_{ji,k})} \\ &= \log\sum_{k=1}^{Q}\exp(\mathbf{S}_{ji,k}) - \mathbf{S}_{ji,j}.\end{aligned} \tag{7}$$

And the utterance-level GE2E loss $\mathcal{L}_u$ is the sum of all speaker embedding losses over the similarity matrix, shown as:

$$\mathcal{L}_u(\mathbf{x};\mathbf{w}) = \sum_{j,i}\mathcal{L}(\tilde{\boldsymbol{e}}_{ji}). \tag{8}$$

For the text-independent speaker verification, each extracted segment-level embedding is expected to capture the speaker characteristics. In order to guide the segment attention to focus on the segments with more speaker discriminations, we further incorporate the similarity loss of segment-level embeddings. The segment-level GE2E loss $\mathcal{L}_s$ is similar to the utterance-level GE2E loss $\mathcal{L}_u$ except that it takes all segment-level embeddings as input, which could help the proposed model to learn more effective ways of embedding fusion and accelerate model convergence. The objective function can be formulated as:

$$\mathcal{L}_s(\mathbf{x};\mathbf{w}) = \sum_{j,i}\sum_{n}\mathcal{L}(\mathbf{e}_{ji,n}). \tag{9}$$

Finally, the utterance-level GE2E loss, segment-level GE2E loss and penalty loss are combined together to construct the total loss, shown as:

$$\mathcal{L} = \mathcal{L}_u + \lambda_s\mathcal{L}_s + \lambda_p\mathcal{L}_p \tag{10}$$

The magnitude of the segment-level GE2E loss and penalty loss is controlled by hyperparameters λ_s and λ_p. With the joint optimization of the segment-level and utterance-level GE2E loss, both local details and global information of utterances are taken into account. Our proposed method can extract the unified speaker embeddings for utterances of variable duration, which unifies the process of training and testing and avoids the mismatch between them.

IV. Experiments

We systemically evaluate the speaker verification performance on DiDi Speaker Dataset, Tongdun and VoxCeleb [19] corpora. The proposed deep segment attentive embedding is compared with the generalized end-to-end loss based embedding as well as the traditional i-vector. We use Equal Error Rate (EER) to quantify the system performance.

A. Data

DiDi Speaker Dataset. The DiDi Speaker Dataset is a large-scale speaker verification corpus, which contains more than 1.8M utterances from 200K Chinese speakers in training set. For evaluation, we use an additional 1.5K speakers with 3 enrollment utterances and 7.2 evaluation utterances per speaker.

Tongdun. The corpus is from the speaker verification competition held by Tongdun technology company [20], which consists of more than 120K utterances from $1,500$ Chinese speakers in training set and $3,000$ trial pairs in test set. Most of the training data are short utterances with average duration of 3.7s, while utterances in test set are very long and average duration is 20s.

VoxCeleb. The training set consists of more than 140K utterances of $1,251$ speakers. And $37,720$ trial pairs from 40 speakers are used as evaluation data for the verification process.

B. i-vector system

The i-vector system uses 20-dimensional MFCCs with their first and second derivatives as front-end features. Cepstral mean normalization is applied. An i-vector of 400 dimensions is then extracted from the acoustic features using a 2048-mixture UBM and a total variability matrix. Mean subtraction, whitening, and length normalization [21] are applied to the i-vector as preprocessing steps, and the similarity is measured using a PLDA model with a speaker space of 400 dimensions.

C. Deep speaker embedding system

For deep speaker embedding systems, we take the 40-dimensional filter-banks with 32-ms Hamming window and 16-ms frame shift as the input features, and each dimension of features is normalized to have zero mean and unit variance. A combination of 3-layer LSTM and a linear projection layer is used to extract the speaker embeddings. Each LSTM layer contains 512 nodes, and the linear projection layer is connected to the last LSTM layer, whose output size is 256. Therefore, we can extract 256-dimension speaker embeddings according to the outputs of the linear projection layer. The cosine similarity score of the pair of embedding vectors is computed to verify the speaker. According to [7], the scaling factors w and b in Eq. (6) are initialized to 10 and 5, respectively.

We take the LSTM-based speaker embedding system proposed by Wan [7] as the baseline, which is optimized by GE2E loss. Let us denote the baseline system as "LSTM-GE2E". "LSTM-GE2E" uses the local segments truncated from utterances to learn the utterance-level speaker embedding. The length of segments is randomly chosen within $[120, 160]$, but the length in a batch is fixed. In the testing phase, each utterance is segmented by a sliding window of 140 frames with 50% overlap. We extract the embedding of each segment and then average them as the speaker embedding of the utterance. The embedding of each segment is obtained by performing a frame-level attention pooling operator on the outputs of the linear projection layer. Note that we also use the last frame of outputs as the segment embedding, but the performance is slightly worse than the attention pooling operation.

Compared to "LSTM-GE2E", the proposed deep segment attentive embedding system uses the whole utterance to learn the utterance-level speaker embedding by the segment attention, which is denoted as "DSAE-GE2E". The segment attention is implemented by performing the multi-head attention pooling on the segment-level embeddings. The attention dim d_a is set to 128 and the attention head number d_r is chosen from $[1, 3, 5]$. In addition, "DSAE-GE2E" is jointly optimized by the utterance-level and segment-level GE2E losses, as shown in Eq. (10). The weights λ_s and λ_p of terms in Eq. (10) are experimentally set to 0.2 and 0.001, respectively.

All deep speaker embedding models are trained from a random initialization by an Adam optimizer [22]. The initial learning rate is set to 0.001 and decayed according to the performance of the validation set. For each batch training, we randomly choose 640 utterances of 64 speakers with 10 utterances per speaker. We mention that the length of segments in a batch is fixed. About 30M batches are used to train the network. In addition, the L_2 norm of gradient is clipped at 3 [23].

TABLE I
SPEAKER VERIFICATION RESULTS ON TONGDUN AND VOXCELEB.

Embedding	EER on Tongdun	EER on VoxCeleb
i-vector	3.0	8.9
LSTM-GE2E	2.0	6.2
DSAE-GE2E-1	1.5	5.8
DSAE-GE2E-3	1.3	5.5
DSAE-GE2E-5	1.0	5.2

D. Results

In the following results, "LSTM-GE2E" refers to the speaker embedding system trained with GE2E loss. "DSAE-GE2E-k" denotes the proposed deep segment attentive embedding system with the multi-head attention layer of k attention heads.

The second column of the Table I shows the performance on Tongdun test set. All deep learning based speaker embedding systems outperform the traditional i-vector system, which shows the effectiveness of the deep speaker embeddings. In general, the proposed "DSAE-GE2E" consistently and significantly outperform "LSTM-GE2E". For the multi-head attention layer, more attention heads achieve greater improvement. "DSAE-GE2E-1" is 25% better in EER than "LSTM-GE2E" and "DSAE-GE2E-5" outperform "LSTM-GE2E" by 50%.

The performance on VoxCeleb test set is shown in the last column of Table I. Our proposed "DSAE-GE2E" also outperforms the i-vector system and "LSTM-GE2E", which demonstrates the effectiveness of the proposed method. "DSAE-GE2E-1" is 6.5% better in EER than "LSTM-GE2E" and "DSAE-GE2E-5" outperform "LSTM-GE2E" by 16.1%. The relative EER reduction is smaller than Tongdun corpus because there is little duration difference between VoxCeleb training and testing utterances.

The performance on DiDi Speaker Dataset test set is shown in Table II. Our proposed "DSAE-GE2E" also outperforms "LSTM-GE2E", which demonstrates the effectiveness of the proposed method in the large-scale corpus. "DSAE-GE2E-3" is 18.3% better in EER than "LSTM-GE2E".

Fig. 2 shows the visualization results using t-SNE[24]. We choose two utterances from different speakers and calculate

TABLE II
SPEAKER VERIFICATION RESULTS ON DIDI SPEAKER DATASET.

Embedding	EER (%)
LSTM-GE2E	4.0
DSAE-GE2E-3	3.27

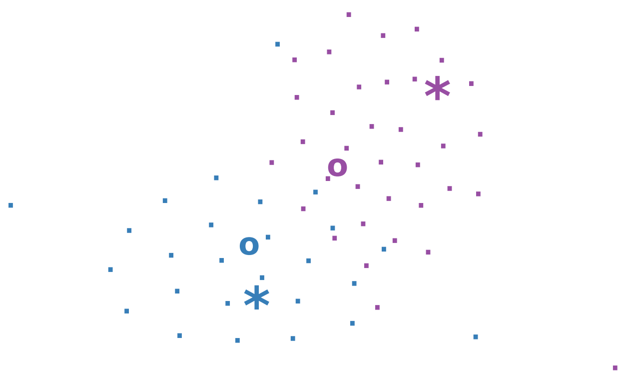

Fig. 2. Visualization using t-SNE. We choose two different speakers and calculate the embedding of each segment. '.' denotes each segment embedding, '*' is results of our proposed "DSAE" and 'o' is results of averaging the segment embeddings. Different colors refer to different speakers.

the segment embedding for each utterance. Compared with the method of averaging each segment embedding, our proposed "DSAE-GE2E" system can better extract speaker discriminations for the given utterance and take into account both local details and global information of utterances.

V. CONCLUSIONS

In this paper, we propose the deep segment attentive embedding method to learn the unified speaker embeddings for utterances of variable duration. Each utterance is segmented by a sliding window and LSTM is used to extract the embedding of each segment. We learn the utterance-level embedding by applying an attentive pooling to all segments embeddings. Moreover, the similarity loss of segment-level embeddings is introduced and jointly optimized with the utterance-level embeddings loss. Experiments on DiDi Speaker Dataset, Tongdun and VoxCeleb demonstrate effectiveness of the proposed method. In future, we will investigate different neural network architectures and attention strategies to obtain greater performance improvement.

VI. ACKNOWLEDGEMENTS

This work was supported in part by the National Natural Science Foundation of China (no. 61573357, no. 61503382, no. 61403370, no. 61273267, no. 91120303) and Project of Xueersi Online School(no. Y9D2M10101).

REFERENCES

[1] N. Dehak, P. J. 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, no. 4, pp. 788–798, 2011.

[2] S. J. D. Prince and J. H. Elder, "Probabilistic linear discriminant analysis for inferences about identity," in *IEEE International Conference on Computer Vision*, 2007, pp. 1–8.

[3] S. Cumani, O. Plchot, and P. Laface, "Probabilistic linear discriminant analysis of i-vector posterior distributions," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2013, pp. 7644–7648.

[4] 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*, 2017.

[5] D. Snyder, D. Garcia-Romero, D. Povey, and S. Khudanpur, "Deep neural network embeddings for text-independent speaker verification," in *INTERSPEECH*, 2017, pp. 999–1003.

[6] E. Variani, X. Lei, E. Mcdermott, I. L. Moreno, and J. Gonzalez-Dominguez, "Deep neural networks for small footprint text-dependent speaker verification," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2014, pp. 4052–4056.

[7] L. Wan, Q. Wang, A. Papir, and I. L. 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.

[8] C. Zhang and K. Koishida, "End-to-end text-independent speaker verification with triplet loss on short utterances," in *Proc. of Interspeech*, 2017.

[9] T. N. Sainath, O. Vinyals, A. Senior, and H. Sak, "Convolutional, long short-term memory, fully connected deep neural networks," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2015, pp. 4580–4584.

[10] G. Heigold, I. Moreno, S. Bengio, and N. Shazeer, "End-to-end text-dependent speaker verification," *Computer Science*, pp. 5115–5119, 2015.

[11] K. Okabe, T. Koshinaka, and K. Shinoda, "Attentive statistics pooling for deep speaker embedding," 2018.

[12] Y. Zhu, T. Ko, D. Snyder, B. Mak, and D. Povey, "Self-attentive speaker embeddings for text-independent speaker verification," *Proc. Interspeech 2018*, pp. 3573–3577, 2018.

[13] P. Kenny, T. Stafylakis, P. Ouellet, M. J. Alam, and P. Dumouchel, "Plda for speaker verification with utterances of arbitrary duration," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2013, pp. 7649–7653.

[14] D. S. Z. L. Na Li, Deyi Tuo and D. Yu, "Deep discriminative embeddings for duration robust speaker verification," in *INTERSPEECH*, 2018, pp. 2262–2266.

[15] J.-M. O. Nam Le, "Robust and discriminative speaker embedding via intra-class distance variance regularization," in *INTERSPEECH*, 2018, pp. 2257–2261.

[16] A. R. Sarthak Yadav, "Learning discriminative features for speaker identification and verification," in *INTERSPEECH*, 2018, pp. 2237–2241.

[17] Z. Lin, M. Feng, C. N. d. Santos, M. Yu, B. Xiang, B. Zhou, and Y. Bengio, "A structured self-attentive sentence embedding," *arXiv preprint arXiv:1703.03130*, 2017.

[18] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th international conference on machine learning (ICML-10)*, 2010, pp. 807–814.

[19] A. Nagrani, J. S. Chung, and A. Zisserman, "Voxceleb: a large-scale speaker identification dataset," *arXiv preprint arXiv:1706.08612*, 2017.

[20] "Tongdun Technology Speaker Verification Competition," https://www.kesci.com/home/competition/5b4eb2cfe87957000f9024a4/.

[21] D. Garcia-Romero and C. Y. Espy-Wilson, "Analysis of i-vector length normalization in speaker recognition systems," in *INTERSPEECH 2011, Conference of the International Speech Communication Association, Florence, Italy, August*, 2011, pp. 249–252.

[22] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[23] R. Pascanu, T. Mikolov, and Y. Bengio, "Understanding the exploding gradient problem," *CoRR, abs/1211.5063*, 2012.

[24] L. v. d. Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of machine learning research*, vol. 9, no. Nov, pp. 2579–2605, 2008.