Interspeech 2018
2-6 September 2018, Hyderabad

End-to-end text-dependent speaker verification using novel distance measures

Subhadeep Dey[1,2], *Srikanth Madikeri*[1], *and Petr Motlicek*[1]

[1]Idiap Research Institute, Martigny, Switzerland
[2]Ecole Polytechnique Fédérale de Lausanne, Lausanne, Switzerland
subhadeep.dey@idiap.ch, srikanth.madikeri@idiap.ch, petr.motlicek@idiap.ch

Abstract

This paper explores novel ideas in building end-to-end deep neural network (DNN) based text-dependent speaker verification (SV) system. The baseline approach consists of mapping a variable length speech segment to a fixed dimensional speaker vector by estimating the mean of hidden representations in DNN structure. The distance between two utterances is obtained by computing L2 norm between the vectors. This approach performs worse than the conventional Gaussian Mixture Model-Universal Background Model (GMM-UBM) based SV on a publicly available corpora. We believe that a degraded performance is due to the employed averaging operation, which may not capture the phonetic information of an utterance. Recent studies indicate that techniques exploiting phonetic information in addition to speaker is beneficial for this task. This paper therefore proposes to incorporate content information of the speech signal by computing distance function with linguistic units co-occuring between enrollment and test data. The whole network is optimized by employing a triplet-loss objective in an end-to-end fashion to estimate SV scores. Experiments on the RSR2015 dataset indicate that the proposed approach outperforms GMM-UBM system by 48% and 36% relative equal error rate for fixed-phrase and random-digit conditions respectively.

Index Terms: speaker verification, speaker embedding, deep neural network, end-to-end speaker verification, i-vector

1. Introduction

In the last decade, the i-vector approach has shown to provide state-of-the-art speaker verification (SV) results [1, 2, 3]. The low error rates of SV make it attractive for potential applications. They require the system to perform verification on short audio recordings with a specific content being spoken by the speaker. This mode of authentication is referred to as text-dependent SV [4, 5, 6].

Text-dependent SV can be implemented in various ways [7, 6, 8, 9, 10] (such as phrase, seen-content, random-digit, or short commands - based authentication). In this paper, we are interested in fixed-phrase and random-digit type of text-dependent SV systems. In case of fixed-phrase SV, the phrase chosen by the user during the enrollment phase has to match the test phrase. As implemented in RSR2015 dataset, for random-digit task, the speaker utters a random sequence of ten unique digits during enrollment phase while in test, the system prompts the user to speak a permutation of five distinct digits. In this paper, it is assumed that the user has pronounced the content of the test data correctly [7, 6].

The traditional approaches to address the text-dependent SV involve Gaussian Mixture Model-Universal Background Model (GMM-UBM), i-vector, or Joint Factor Analysis (JFA) [8, 11, 7]. In [12], the parameters of the i-vector model were estimated by conditioning on the content of the speech signal. A back-end classifier was further trained by concatenating i-vectors corresponding to each phonetic units. In [13], JFA was trained using the features corresponding to segmented digits as input. A significant gain in performance was observed compared to the baseline system.

Recently, deep neural network (DNN) based speaker modeling has shown to provide performance comparable to the state-of-the-art i-vector system [14, 15, 16, 17]. In this case, it assumes that the hidden representations of the DNN are sufficient to discriminate among speakers. In literature, the DNN based speaker classification can be performed either using, (i) minimizing classification loss, or (ii) end-to-end distance loss [14, 16, 18].

In [19], speaker classification was performed by employing a DNN with a final soft-max layer representing the speaker classes. During evaluation, the final layer is discarded and the hidden representations for each frame of the utterance are accumulated to obtain a speaker template. This representation is also referred to as "d-vector". The d-vectors extracted from the enrollment and the test utterances are finally compared to provide SV scores. Results obtained using this approach are comparable to the baseline system on their proprietary "Ok Google" data set.

The end-to-end approach involves training the network based on similarity score of a pair of audio recordings [18, 17]. We consider triplet-loss as the objective function for training such an end-to-end system [20, 21]. Recently, triplet-loss has shown to be successful for SV [17]. It involves optimization based on pairwise distance between same-speaker and different-speaker. This technique can be used to obtain both speaker embedding and output SV scores directly. This approach consists of obtaining a fixed dimensional speaker vector for an utterance by accumulating the hidden representation of DNN. The similarity between two utterances is measured using Euclidean-distance metric. Such systems have not shown to outperform the state-of-the-art i-vector system on a publicly available dataset [22]. We hypothesize that the degraded performance is due to the averaging operation which may ignore the content information of the speech signal. In the past, it has been shown that performance of text-dependent SV can be substantially improved by exploiting phonetic information of an utterance [6, 12]. This paper aims to incorporate this information in the DNN training by computing the distance between the enrollment and test data using the common linguistic units in an unsupervised way (i.e. without using the text transcript). The objective function is designed to selectively attend to parts of the enrollment utterance for producing SV scores. Experiments were done on the RSR2015 dataset [7] and reveal significant improvements in SV performance over the baseline for both fixed-phrase and random-digit conditions.

The paper is organized as follows. Section 2 describes the

10.21437/Interspeech.2018-2300

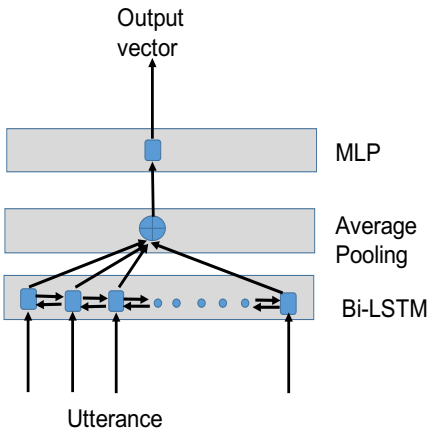

Figure 1: *The neural network architecture of triplet-loss approach for text-dependent SV.*

traditional SV system considered in this paper, while Section 3 describes the DNN based speaker modelling techniques. Sections 4 and 5 present the proposed distance measures and experimental setup for evaluating the system. We discuss the achieved results in Section 6 and finally, the paper is concluded in Section 7.

2. Baseline System

The traditional SV system relies on a factor analysis to provide high performance. In this approach, any speech utterance is represented by a fixed dimensional vector, also referred to as the identity vector (or i-vector) [1, 2, 3]. It involves the projection of a high dimensional mean supervector ($\mathbf{m}$) to a low dimensional vector (usually of size 400), as given by the following equation

$$\mathbf{m} = \mu + \mathbf{T}\mathbf{w}, \tag{1}$$

where $\mathbf{T}$ is referred to as total variability matrix, $\mathbf{w}$ is the i-vector of the utterance and μ is the UBM mean supervector.

3. DNN based speaker modelling

The i-vector system described above assumes the speaker data to be generated from a GMM [1, 2]. In another direction, modelling speakers with a DNN has shown to be beneficial for SV [19, 17]. On some datasets, the DNN based speaker modelling has shown to outperfom the traditional i-vector system [23, 16]. This paper explores such an end-to-end SV system employing triplet loss function. The loss aims at minimizing the intra-speaker distance and maximizing inter-speaker distance simultaneously. This approach has shown to provide low error rates in variety of machine learning tasks, including image processing [21, 20]. Recently, this technique has shown promising results on one of SV task as well [17].

3.1. Triplet-loss

The triplet-loss consists of three utterances (also referred to as the triplet, τ), represented by the set $\{\mathbf{U}^a, \mathbf{U}^p, \mathbf{U}^n\}$, an used as an input for training the network. In literature, these examples are popularly referred to as the anchor, positive and negative instances [21, 20]. These triplet utterances are selected in such a way that the anchor and positive utterances belong to the same class while the anchor and negative examples do not share the same speaker identity. Assuming the hidden representation of the utterance ($\mathbf{U}$) is represented by the function $\mathbf{f}(\mathbf{U})$, the triplet loss (L) is given by

$$\mathrm{L}(\mathbf{U}^a, \mathbf{U}^p, \mathbf{U}^n) = \mathrm{d}(\mathbf{f}(\mathbf{U}^a), \mathbf{f}(\mathbf{U}^p)) - \mathrm{d}(\mathbf{f}(\mathbf{U}^a), \mathbf{f}(\mathbf{U}^n)) + \alpha, \tag{2}$$

where $\mathrm{d}(.)$ is the function that computes the distance between two vectors, and α is a predefined constant (0.1 is used in our experiments). The most commonly used distance functions are Euclidean and cosine similarity [21, 20]. In this paper, we used Euclidean distance.

We apply the same network topology as used in speaker diarization and speaker verification [24] (as shown in Figure 1). The input is fed to a bi-directional Long Short Term Memory (bi-LSTM) with $\tanh$ activation function to produce speaker representation of a speech frame [24, 18]. This output is fed to the Average Pooling layer that estimates the mean of the activations to produce a vector [15, 24]. The speaker embedding or vector is then forwarded to a fully connected (FC) layer.

3.2. Triplet-loss with attention

In this work, we also explore an extension of the triplet-loss network by applying an attention mechanism. This technique has also been used in the work to train a Siamese network [16]. The network architecture is shown in Figure 2. Unlike the conventional triplet network (as described above), the Average Pooling layer (in Figure 2) provides a speaker representation ($\mathbf{h}^{'}$) by linearly combining the hidden activations (denoted by $\{\mathbf{h}_1, \mathbf{h}_2, \cdots, \mathbf{h}_M\}$) after the first layer (bi-LSTM) and is given by the following equation

$$\mathbf{h}^{'} = \sum_{i=0}^{M} \mathrm{w_i}\mathbf{h}_i,$$

where $\mathrm{w_i}$ are the weights of the i^{th} speech frames. The weights are computed by using a FC (denoted by the function g, the first FC layer of Figure 2) and a $\tanh$ activation function as follows

$$\mathrm{w_i} = \tanh(\mathrm{g}(\mathbf{h}_i)),$$

and finally the weights are normalized over an utterance to obtain the attention vector as follows

$$\mathrm{w_i} = \frac{\mathrm{w_i}}{\sum_j \mathrm{w_j}}.$$

The attention based speaker embedding ($\mathbf{h}^{'}$) is then used for training the triplet-loss as given by Equation 2.

4. Distance function for DNN

In Section 3, speaker embedding of an utterance is obtained by computing the mean or weighted mean of the hidden activations in DNN. The distance between two utterances is computed as the Euclidean distance between speaker vectors. However, this strategy may not use the phonetic content of the speech signal. In the past, it has been shown that employing content information of an utterance in addition to speaker is beneficial for text-dependent SV [6, 12]. In this section, we explore a new distance function that exploits phonetic information of the speech signal implicitly.

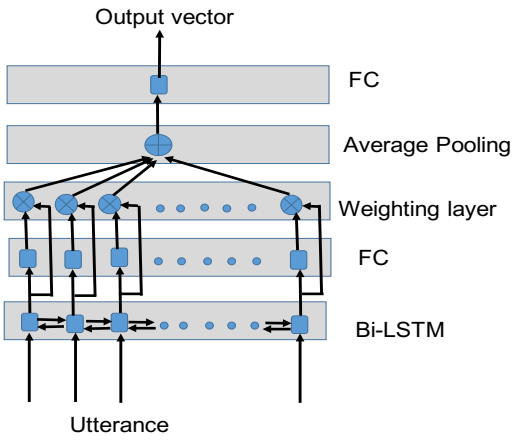

Figure 2: *The neural network architecture of triplet-loss approach with attention mechanism for text-dependent SV.*

4.1. Proposed end-to-end distance functions

For the proposed distance functions, the network architecture is similar to that of the triplet loss network (Figure 1). The main difference is that the Average Pooling layer has been removed from the network. Thus, an utterance produces as many hidden speaker embeddings as the number of speech frames. Let us assume the two utterances ($\mathbf{H}_e$ and $\mathbf{H}_t$) produce the following hidden representations $\{\mathbf{h}_{e,1}, \mathbf{h}_{e,2}, \mathbf{h}_{e,3}, \cdots, \mathbf{h}_{e,i}, \cdots, \mathbf{h}_{e,R}\}$ and $\{\mathbf{h}_{t,1}, \mathbf{h}_{t,2}, \mathbf{h}_{t,3}, \cdots, \mathbf{h}_{t,j}, \cdots, \mathbf{h}_{t,C}\}$. In this paper, we explore three distance functions as described below:

- **Average distance**: The average distance (D) between two utterances $\mathbf{H}_e$ and $\mathbf{H}_t$ is given by the following equation

$$\mathrm{D}(\mathbf{H}_e, \mathbf{H}_t) = \frac{1}{RC} \sum_{i,j} \mathrm{d}(\mathbf{h}_{e,i}, \mathbf{h}_{t,j}),$$

  where d is Euclidean distance between two vectors ($\mathbf{h}_{e,i}$ and $\mathbf{h}_{t,j}$). It is to be noted that if the cosine-distance is used as d(.) , then through some algebraic manipulation, it can be shown that the average distance (D) is similar to the conventional triplet loss function given in Equation 2.

- **Minimum distance**: The next similarity measure that we consider is based on scoring using the common set of phones between two utterances. Assuming that the hidden representation of a speech frame contains phonetic information as well, the phonetic information co-occuring between the pair of utterances is obtained as follows

$$\mathrm{D}(\mathbf{H}_e, \mathbf{H}_t) = \frac{1}{C} \sum_{j} min_i \mathrm{d}(\mathbf{h}_{e,i}, \mathbf{h}_{t,j}). \quad (3)$$

  This type of distance function has been used in our previous work (in the GMM framework) but mainly as a post-processing step [6]. It is to be noted that this proposed distance is not symmetric. The minimum function finds the closest match of the an utterance with hidden representation $\mathbf{h}_{t,j}$ against other features in $\mathbf{H}_t$.

- **Attention based distance function**: The previous distance function assumes equal amount of the information are captured by each hidden representation in $\mathbf{H}_t$. The following loss function is proposed to better model this imbalance

$$\mathrm{D}(\mathbf{H_e}, \mathbf{H_t}) = \sum_{j} \mathrm{w}_j min_i \mathrm{d}(\mathbf{h}_{e,i}, \mathbf{h}_{t,j}),$$

  where w_j is the weight of the j^{th} hidden representation. The network for performing this optimization is similar to the one described in Section 3.2, with the difference being in the distance function. We train the network using Equation 2 by replacing d with the proposed distance function D.

The networks described above can be directly used in an end-to-end training to produce SV scores by applying the appropriate distance function.

4.2. Distance function using PLDA

It has been shown that applying a back-end classifier to post-process the scores is beneficial for SV [14]. In this work, we apply Probabilistic Linear Discriminant Analysis (PLDA) model to compute the distance function d(.) as well. The PLDA is trained on these hidden representations $\mathbf{H}_e$ and $\mathbf{H}_t$ by using the speaker labels for training. The PLDA based scoring is applied only during evaluation. The training of the network is done as described in the previous section (Section 4.1).

5. Experimental Setup

In this section, experimental setup of the baseline and the proposed systems are described.

5.1. Evaluation and Training Data

For the fixed-phrase and random-digit tasks, we use the *dev* and *bgnd* portions of RSR 2015 from Part 1, 2 and 3 as the training data [7]. This part consists of 61K utterances spoken by 94 speakers. The experiments were performed on the female speaker subset only. We now describe the evaluation data for the fixed-phrase and random-digit tasks.

- **Fixed-phrase**: Experiments are conducted on the Part 1 of RSR 2015 for the phrase based text-dependent SV [7, 4, 5]. This part comprises 49 speakers uttering 30 phrases. The systems are evaluated in speaker-mismatch condition only. It consists of 8'810 target and 422'880 impostor trials.
- **Random-digit**: We performed experiments on the Part 3 of RSR2015 dataset to evaluate our systems. It consists of 49 speaker uttering a permutation of digits. The enrollment data are represented by speakers pronouncing ten digits while the test utterances consist of five (randomply selected) digits. The average duration of the enrollment utterance is 6 s while the test is 3 s. The total number of target and impostor trials is 5'283 and 253'584 respectively.

5.2. Features and i-vector system

The front-end SV system extracts Mel Frequency Cepstral Coefficients (MFCC) of 20 dimensions from 25 ms frames of speech signal with 10 ms sliding window with the delta and double delta features appended to it. 512 mixture Universal Background Model (UBM) is trained followed by 200 dimensional i-vector extractor. Finally, a PLDA is trained, as part of the standard recipe of text-independent SV system, with speaker labels provided by training data.

Table 1: *Performance of the various systems in terms of EER (%) on RSR2015 for fixed-phrase and random-digit tasks. The **GMM-UBM** system performs the best among all the baseline approaches.*

Systems	Fixed-phrase (%)	Random-digit
i-vector	4.3	11.8
GMM-UBM	**2.3**	**7.8**
Triplet	6.9	15.3
Triplet-Attn	4.4	11.7

Table 2: *Performance of the various proposed systems in terms of EER(%) on RSR2015 for fixed-phrase and random-digit tasks. The systems are evaluated using end-to-end objective function.*

Systems	Fixed-phrase (%)	Random-digit
Avg-Dist	11.2	29.7
Min-Dist	1.8	**7.6**
Attn-Dist	9.4	29.1

Table 3: *Performance of the various systems in terms of EER(%) on RSR2015 fixed-phrase and random-digit tasks. The proposed systems are evaluated using a back-end PLDA classifier.*

Systems	Fixed-phrase (%)	Random-digit
Avg-Dist	3.4	15.7
Min-Dist	**1.2**	**5.0**
Attn-Dist	1.4	5.4

5.3. Triplet loss network

We use online triplet mining approach as described in [20] to select training examples. For each epoch, we generate triplets ($\mathbf{U}^a$, $\mathbf{U}^p$, $\mathbf{U}^n$) such that the phonetic content of these utterances has maximal overlap. We create a total of 300K triplets per epochs. A learning rate of 0.001 was used throughout the experiments with "RMS-prop" as the optimizer. The triplet network uses 400 dimensional hidden representation. Pytorch was used for performing the experiments [25]. The performances of various systems are reported in terms of EER.

6. Experimental Results and Discussions

This sections presents speaker verification results obtained with the baseline and the proposed systems. We evaluated the performance of the following systems on the random-digit and fixed-phrase tasks:

- **i-vector**: Conventional i-vector system using Gaussian Mixture Model (GMM). PLDA is trained as a back-end classifier.
- **GMM-UBM**: GMM-UBM system built by pooling data from all speakers. The speaker models are obtained using maximum-a-posteriori adaptation.
- **Triplet**: This approach optimizes the triplet-loss function on three utterances. The conventional approach to using triplet-loss network is described in Section 3.1. This technique employs a bi-LSTM and a FC layer. Speaker model (or representation) of an utterance is obtained by collecting the activations after the Average Pooling layer (see Figure 1). PLDA is further trained on these representation to obtain SV scores. This system will be referred to as **Triplet**. The approach applying attention based mechanism and triplet loss (as described in Section 3.2) is referred to as **Triplet-Attn**.
- **Proposed systems**: The triplet-loss networks applying the average, minimum and attention-based distance are referred to as **Avg-Dist**, **Min-Dist** and **Attn-Dist** (as described in Section 4) respectively. The systems are evaluated using end-to-end objective function as described in Section 4.1. We also evaluate the performance of these proposed systems by applying PLDA as a post-processing step as described in Section 4.2.

6.1. Baseline SV systems

Table 1 shows the performance of **i-vector**, **GMM-UBM**, **Triplet** and **Triplet-Attn** based systems. The performance of the baseline **i-vector** and **GMM-UBM** systems is comparable to the results published by others [6]. From Table 1, it can be observed that **GMM-UBM** significantly outperforms **i-vector**, which is consistent with the published results. The results presented for **GMM-UBM** system are obtained after applying T-norm.

For the triplet based network, the **Triplet-Attn** outperforms **Triplet** system. This suggests the importance of attention in producing the speaker representation of an utterance. Furthermore, **Triplet-Attn** outperforms the **i-vector** by a 0.1% absolute EER for random-digit task. However, both the triplet based approaches (**Triplet-Attn** and **Triplet**) perform worse than the baseline **GMM-UBM**.

6.2. Proposed approaches

Table 2 shows the performance of the proposed SV systems evaluated against their respective end-to-end objective functions. The results suggest that the **Min-Dist** performs the best among all the system with relative 21.7% (2.3% to 1.8% absolute) and 2.6% (7.8% to 7.6% absolute) improvement in EER for fixed-phrase and random-digit tasks respectively over the **GMM-UBM**. The performance of **Avg-Dist** and **Attn-Dist** systems is worse than of **Min-Dist**.

We also investigate the use of PLDA (as described in Section 4.2) to produce SV scores. Table 3 shows the performance. We observe that all the systems get improved when back-end classifier is exploited on top of the hidden DNN representations (speaker embedding). **Attn-Dist** benefits the most with 81% relative (29.1% to 5.4% absolute) EER and outperforms the baseline **GMM-UBM** by 31% relative (7.8% vs 5.4% absolute) EER. Futhermore, **Min-Dist** provides the best EER of 1.2% and 5.0% for fixed-phrase and random-digit tasks respectively. The results of the proposed systems do not necessarily improve by applying T-norm on the scores. In future, we will investigate normalization techniques to be applied for the proposed system.

7. Conclusions

In this paper, we explored end-to-end based text-dependent SV system using novel distance measures. The DNN based network employing such a distance is optimized using triplet-loss based objective function. The similarity value is computed by using the words or phonetic units co-occuring between the enrollment and test data. The proposed approach is designed to attend to relevant parts of the enrollment by selecting closest region of the test utterance. Experiments on the RSR2015 corpora show that the proposed technique outperforms the baseline GMM-UBM based baseline system by 48% and 36% relative EER for fixed-phrase and random-digit based text-dependent SV.

8. References

[1] O. Glembek *et al.*, "Simplification and optimization of i-vector extraction." In Proc. of ICASSP, 2011, pp. 4516–4519.

[2] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 19, no. 4, pp. 788–798, May 2011.

[3] D. G. Romero and A. McCree, "Supervised domain adaptation for i-vector based speaker recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2014, Florence, Italy, May 4-9, 2014*, 2014, pp. 4047–4051.

[4] A. Larcher, K. A. Lee, B. Ma, and H. Li, "Modelling the alternative hypothesis for text-dependent speaker verification," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, May 2014, pp. 734–738.

[5] A. Larcher, K. Lee, B. Ma, and H. Li, "Phonetically-constrained plda modeling for text-dependent speaker verification with multiple short utterances," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, May 2013, pp. 7673–7677.

[6] S. Dey, S. Madikeri, P. Motlicek, and M. Ferras, "Content normalization for text-dependent speaker verification," *Proc. Interspeech 2017*, pp. 1482–1486, 2017.

[7] A. Larcher, K. Lee, B. Ma, and H. Li, "Text-dependent speaker verification: Classifiers, databases and RSR2015," *Speech Communication*, vol. 60, pp. 56–77, 2014. [Online]. Available: http://dx.doi.org/10.1016/j.specom.2014.03.001

[8] N. Scheffer and Y. Lei, "Content matching for short duration speaker recognition." in *INTERSPEECH*, 2014, pp. 1317–1321.

[9] S. Dey, P. Motlicek, S. Madikeri, and M. Ferras, "Exploiting sequence information for text-dependent speaker verification," in *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*. Ieee, 2017, pp. 5370–5374.

[10] S. De, P. Motlicek, S. Madikeri, and M. Ferras, "Template matching for text-dependent speaker verification," *Speech Communication*, vol. 88, pp. 96–105, 2017.

[11] P. Kenny, T. Stafylakis, J. Alam, P. Ouellet, and M. Kockmann, "Joint factor analysis for text-dependent speaker verification," *Odyssey*, 2014.

[12] L. Chen, K. A. Lee, B. Ma, W. Guo, H. Li, and L.-R. Dai, "Phone-centric local variability vector for text-constrained speaker verification," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[13] T. Stafylakis, M. J. Alam, and P. Kenny, "Text-dependent speaker recognition with random digit strings," *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 24, no. 7, pp. 1194–1203, 2016.

[14] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust dnn embeddings for speaker recognition," *ICASSP, Calgary*, 2018.

[15] D. Snyder, D. Garcia-Romero, D. Povey, and S. Khudanpur, "Deep neural network embeddings for text-independent speaker verification."

[16] F. Chowdhury, Q. Wang, I. L. Moreno, and L. Wan, "Attention-based models for text-dependent speaker verification," *arXiv preprint arXiv:1710.10470*, 2017.

[17] 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.

[18] G. Heigold, I. Moreno, S. Bengio, and N. Shazeer, "End-to-end text-dependent speaker verification," in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 5115–5119.

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

[20] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 815–823.

[21] E. Hoffer and N. Ailon, "Deep metric learning using triplet network," in *International Workshop on Similarity-Based Pattern Recognition*. Springer, 2015, pp. 84–92.

[22] D. Snyder, P. Ghahremani, D. Povey, D. Garcia-Romero, Y. Carmiel, and S. Khudanpur, "Deep neural network-based speaker embeddings for end-to-end speaker verification," in *Spoken Language Technology Workshop (SLT), 2016 IEEE*. IEEE, 2016, pp. 165–170.

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

[24] H. Bredin, "Tristounet: triplet loss for speaker turn embedding," in *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*. IEEE, 2017, pp. 5430–5434.

[25] Pytorch. (2017) https://github.com/pytorch. [Online]. Available: https://github.com/pytorch