INTERSPEECH 2020
October 25–29, 2020, Shanghai, China

Deep Speaker Embedding with Long Short Term Centroid Learning for Text-independent Speaker Verification

Junyi Peng[1], *Rongzhi Gu*[1], *Yuexian Zou*[1,2*]

[1]ADSPLAB, School of ECE, Peking University, Shenzhen, China
[2]Peng Cheng Laboratory, Shenzhen, China
{pengjy, 1701111335, zouyx}@pku.edu.cn

Abstract

Recently, speaker verification systems using deep neural networks have shown their effectiveness on large scale datasets. The widely used pairwise loss functions only consider the discrimination within a mini-batch data (short-term), while either the speaker identity information or the whole training dataset is not fully exploited. Thus, these pairwise comparisons may suffer from the interferences and variances brought by speaker-unrelated factors. To tackle this problem, we introduce the speaker identity information to form long-term speaker embedding centroids, which are determined by all the speakers in the training set. During the training process, each centroid dynamically accumulates the statistics of all samples belonging to a specific speaker. Since the long-term speaker embedding centroids are associated with a wide range of training samples, these centroids have the potential to be more robust and discriminative. Finally, these centroids are employed to construct a loss function, named long short term speaker loss (LSTSL). The proposed LSTSL constrains that the distances between samples and centroid from the same speaker are compact while those from different speakers are dispersed. Experiments are conducted on VoxCeleb1 and VoxCeleb2. Results on the VoxCeleb1 dataset demonstrate the effectiveness of our proposed LSTSL.

Index Terms: speaker verification, speaker embedding, speaker centroid, x-vectors

1. Introduction

Speaker verification (SV) is the process of automatically verifying an speech utterance whether belongs to a claimed identity. According to the restriction of the uttered content, speaker verification can be approached as a text-dependent speaker verification (TD-SV) or text-independent speaker verification (TI-SV) task [1]. Without the constraint of a specific phrase, TI-SV has a wide variety of applications including smart home and speech monitor.

The combination of i-vector and Probabilistic Linear Discriminant Analysis (PLDA) has dominated for over 10 years [2]. In these systems, i-vector is employed as the feature extractor and PLDA is served as the back-end classifier. These two components are loosely connected and optimized using different criteria.

In the last few years, more studies have presented superior results using deep neural networks for extracting speaker representations. In [3], four fully connected layers are trained for speaker classification in the training step. In the verification step, the speaker embedding (‘d-vector’) is calculated from averaging the last hidden layer’s output over frames. Using this pipeline, more well-designed multi-class classification loss functions such as angular softmax loss (ASoftmax), additive margin softmax loss (AMSoftmax), additive angular margin softmax (ArcSoftmax) and large margin cosine loss (LMCL) have been proposed for SV task [4, 5, 6, 7]. However, these systems optimized by softmax loss and its variants do not take the discrimination of different speaker embedding pairs into consideration, leading to difficulty in distinguishing between positive pair (verification utterances belong to the same speaker) and negative pair (verification utterances belong to different speakers).

To address this problem, some efforts have been made to explore end-to-end speaker verification models. These methods drive the network to directly discriminate the positive pair and negative pair. In an end-to-end system, the loss function plays a significant role in minimizing intra-speaker divergence and maximizing inter-speaker separability of the speaker embeddings. Following this idea, several metric learning methods such as contrastive loss [8, 9], triplet loss [10, 11] are applied to directly optimize the speaker representation. The main concept behind these methods is to construct training pairs or triplets to simulate the enrollment and testing stages of speaker verification. Nevertheless, these methods suffer from dramatic data expansion when constituting sample pairs during the training and the performance is sensitive to the pair sampling strategies. Also, choosing an appropriate sampling strategy for different datasets is difficult and time-consuming.

Very recently, affinity loss (AL) is proposed for short utterance speaker verification [12]. AL does not rely on the pair selection strategy and flexibly makes use of all speaker embeddings pairs’ comparison information in a mini-batch, which has the potential to improve the discrimination of speaker embeddings. However, without explicit identity information, the SV model has to exclude the influence brought by speaker-unrelated factors, such as the channel variation and speaker accents, which may interfere the extraction of intrinsic speaker representation. This variation may lead to the decentralization of intra-speaker samples, thus rendering slow convergence and suboptimal performance.

In this paper, based on our previous work [12], we propose a novel loss function referred as long short term speaker loss (LSTSL) for end-to-end SV, which improves the procedure of individual training pair comparison by an updatable speaker embedding centroid learning algorithm. Specifically, the one-hot speaker label information is leveraged to compute the centroid (mean) of each speaker according to mini-batch composition, named as short-term speaker embedding centroids. Then, the speakers involved in the short-term speaker embeddings centroids are used to update the long-term speaker embedding

*Corresponding author. This paper was supported by Shenzhen Science & Technology Fundamental Research Programs (No: JCYJ20170817160058246 & JCYJ20180507182908274).

http://dx.doi.org/10.21437/Interspeech.2020-2470

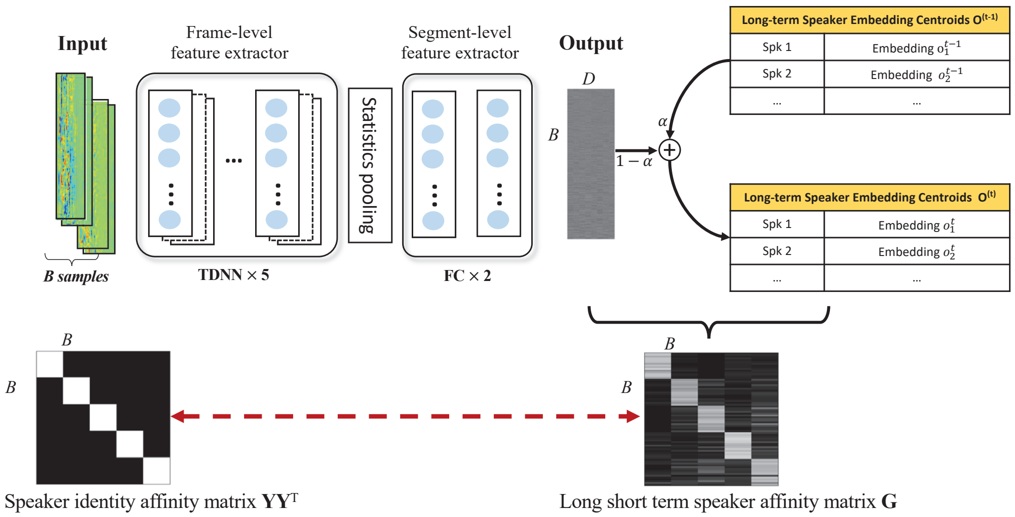
Input
Frame-level feature extractor
Segment-level feature extractor
Output
Statistics pooling
B samples
TDNN × 5
FC × 2
D
B
α
1 − α
Long-term Speaker Embedding Centroids O^(t-1)
Spk 1
Embedding o_1^{t-1}
Spk 2
Embedding o_2^{t-1}
...
...
Long-term Speaker Embedding Centroids O^(t)
Spk 1
Embedding o_1^{t}
Spk 2
Embedding o_2^{t}
...
...
B
B
B
B
Speaker identity affinity matrix $\mathbf{YY}^{\top}$
Long short term speaker affinity matrix $\mathbf{G}$

Figure 1: *The architecture of x-vector with LSTSL. B denotes the batch size of input data, D is the dimension of output speaker embedding. To facilitate understanding, in this figure, the batch of data is sorted by speaker identity. During the training stage, all batches of data are composed randomly.*

centroids. The long-term speaker embedding centroids consider all the speakers in the training set, while the short-term speaker embedding centroids only cover speakers in a mini-batch. Finally, the intra-speaker variability is compressed by minimizing the differences between speaker embeddings and their corresponding updated long-term centroids. The inter-speaker separability is achieved by pulling away the speaker embeddings and updated long-term speaker centroids that belong to different speaker identities. Experimental results on VoxCeleb1 demonstrate the effectiveness of our proposed loss function, and systems optimized by LSTSL outperform existing speaker verification methods and achieve state-of-the-art performance.

The rest of paper is organized as follows. Section 2 gives a brief introduction to the speaker embedding extractor and affinity loss. Section 3 describes the proposed LSTSL in detail. Database description, training paradigm and results analysis are described in Section 4. Section 5 concludes the paper.

2. Related Works

2.1. Speaker Embedding Extractor

The network architecture of our x-vector baseline system is similar to that described in [13], as shown in Fig. 1. The first five TDNN (or 1-dimensional dilated CNN) layers are stacked to extract the frame-level features. The TDNN layers with dilation rates of 2 and 4 are used for the second and third layers, respectively, while the others retain the dilation rate of 1. The kernel sizes of these five layers are 5, 3, 3, 1 and 1, respectively. The final frame-level output vectors of the whole variable-length segment are aggregated into a fixed segment-level vector through the statistics pooling layer. The mean and standard deviation are calculated and then concatenated for statistics pooling. Two additional fully connected layers followed with a softmax layer are used to predict speaker labels.

2.2. Affinity Loss

To better understand the proposed LSTSL, we give a brief review of the original AL. Assume that the output of x-vector feature extractor is $\mathbf{S} = \{\mathbf{s}_b\}_{b=1}^{B} \in \mathbb{R}^{B\times D}$, where B is the mini-batch size and D is the dimension of the output. It is noted that each speaker embedding s_b is a unit-norm speaker embedding (i.e. $\|s\|^2 = 1$). Correspondingly, the speaker identity matrix is $\mathbf{Y} = \{\mathbf{y}_b\}_{b=1}^{B} \in \mathbb{R}^{B\times N}$, where N is the total number of speakers involved in the training set. Following mathematic notations, the matrix $\mathbf{SS}^{\top} \in \mathbb{R}^{B\times B}$ is termed as the speaker embedding affinity matrix and $\mathbf{YY}^{\top} \in \mathbb{R}^{B\times B}$ as speaker identity affinity matrix, respectively. Formally, we define the affinity loss as:

$$\begin{aligned}\mathcal{L}_{\mathcal{AL}} &= \|\mathbf{SS}^{\top} - 2\mathbf{YY}^{\top} + 1\|_F^2 \\ &= \sum_{\substack{i,j \\ \mathbf{y}_i=\mathbf{y}_j}} (1 - cos(\mathbf{s}_i, \mathbf{s}_j))^2 + \sum_{\substack{i,j \\ \mathbf{y}_i\neq\mathbf{y}_j}} (-1 - cos(\mathbf{s}_i, \mathbf{s}_j))^2\end{aligned} \tag{1}$$

where $\|\cdot\|_F^2$ denotes the squared Frobenius norm. It is noted that $(\mathbf{SS}^{\top})_{ij} = cos(\mathbf{s}_i, \mathbf{s}_j)$ indicates the cosine similarity between $\mathbf{s}_i$ and $\mathbf{s}_j$. If segment i and j belong to the same speaker, then the cosine similarity between s_i and s_j should be close to 1. Similarly, $2\mathbf{YY}^{\top} - 1$ is a binary matrix, specifically, if the segment i and j belong to the same speaker (with the same one-hot label vector) then we have $(2\mathbf{YY}^{\top} - 1)_{ij} = 1$. Otherwise, we have $(2\mathbf{YY}^{\top} - 1)_{ij} = -1$.

3. Long short term speaker loss

As discussed in Section 2, AL takes the pair information constructed by speaker labels as supervision. However, without the explicit identity information, only use the pairwise comparison information, the optimization process may be dominated by speaker-unrelated factors, such as genders, channel variations and speaker accents. Thus, the intrinsic speaker representation extraction is hindered and the network may fall into the suboptimal local minima early on in training.

To alleviate this problem, we propose a novel loss function, integrating the one-hot speaker label information and pairwise comparisons as supervision information to optimize the speaker embedding extractor by simultaneously maximizing the separability of speaker embeddings with different identities and the compactness of those with the same identity.

For a randomly composed mini-batch data, we employ a series of matrix operations to compute the speaker embedding centroids. Specifically, considering the speaker identity matrix $\mathbf{Y}$, we use a matrix multiplication $\mathbf{Y}^\top\mathbf{S} \in \mathbb{R}^{N\times D}$ to collect and aggregate the speaker embeddings $\mathbf{S}$ according to the identity. The correlation matrix $\mathbf{Y}^\top\mathbf{Y} \in \mathbb{R}^{N\times N}$ is a diagonal matrix, where each diagonal element denotes the number of occurrences of a specific speaker in a mini-batch. So that the short-term speaker embedding centroids $\mathbf{C}$, which is the speaker centroids in a mini-batch, can be obtained as follow:

$$\mathbf{C} = (\mathbf{Y}^\top\mathbf{Y})^{-1}\mathbf{Y}^\top\mathbf{S} \tag{2}$$

where $\mathbf{C} \in \mathbb{R}^{N\times D}$, each row of $\mathbf{C}$ represents the centroid (mean) speaker embeddings of corresponding speakers in a mini-batch. If a speaker n is not sampled in this mini-batch data, then the speaker's corresponding row vectors is full zeros.

However, $\mathbf{C}$ only considers a limited number of speakers according to mini-batch composition, which is difficult to exclude the influence brought by speaker-unrelated factors. This makes the obtained speaker embedding centroids unstable. To obtain a more precise speaker embedding centroid, instead of the utterances sampled in a mini-batch, we consider all the utterances that belong to a speaker in the training set. Specifically, at each iteration t, we use the short-term speaker embedding centroids $\mathbf{C}^{(t)}$ computed with current batch data, to accumulate and update the long-term speaker embedding centroids $\mathbf{O}^{(t)} = \{o_i^t\}_{i=1}^N \in \mathbb{R}^{N\times D}$, as follows:

$$\mathbf{O}^{(t)} = \alpha * \mathbf{O}^{(t-1)} + (1-\alpha) * \mathbf{C}^{(t)} \tag{3}$$

where a hyper-parameter $\alpha \in [0,1]$ adjusts the ratios between the short-term speaker embedding centroids and the long-term speaker embedding centroids. $\mathbf{O}^{(t-1)}$ denotes the long-term speaker embedding centroids at iteration $t-1$, $\mathbf{O}^{(t)}$ denotes the updated long-term speaker embedding centroids.

To fascinate computation, we broadcast the updated long-term speaker embedding centroids $\mathbf{O}^{(t)}$ as $\mathbf{YC}^{(t)} \in \mathbb{R}^{B\times D}$, where each row vector represents the speaker embedding centroid referred to the one hot label $\mathbf{y}_b$. In this way, we propose the long short term speaker loss (LSTSL) based on the long-term speaker embedding centroids as the following equation:

$$\begin{aligned}\mathcal{L}_{LSTSL} &= \|\mathbf{S}(\mathbf{YO}^{(t)\top} - \mathbf{YY}^\top\|_F^2 \\ &= \|\mathbf{SO}^{(t)\top}\mathbf{Y}^\top - \mathbf{YY}^\top\|_F^2 \\ &= \sum_{\substack{i,j \\ \mathbf{y}_i=\mathbf{y}_j}} (\cos(\mathbf{o}_i^t, \mathbf{s}_j) - 1)^2 + \sum_{\substack{i,j \\ \mathbf{y}_i\neq\mathbf{y}_j}} (\cos(\mathbf{o}_i^t, \mathbf{s}_j) - 0)^2\end{aligned} \tag{4}$$

where $\|\cdot\|_F^2$ denotes the squared Frobenius norm. It is noted that the long short term speaker affinity matrix $\mathbf{G} = \mathbf{SO}^{(t)\top}\mathbf{Y}^\top$ indicates the cosine similarity between speaker embedding centroids and speaker embeddings, as shown in fig 1. Moreover, the first term in Eq. 4 is to increase the intra-speaker compactness. The second term means to pull the cosine similarity between speaker embedding centroids and other speaker embeddings to 0.

Compared with the training pair construction strategy, the updatable speaker embedding centroid learning algorithm iteratively assembles embeddings of all training utterances from a speaker to its centroid. The speaker centroid has the potential to be more robust and discriminative than randomly constructed pairs. Besides, LSTSL additionally constrains the intra-speaker centroid variance using speaker identity information, which makes the optimization process more stable. Compared to the AL [12], LSTSL inherits the advantage of AL that not relies on pair selection strategy, which makes the training process more efficient and convenient. Meanwhile, LSTSL balances the short-term centroids and long-term centroids by a controllable weight. In this way, the weight of the speaker centroids that learned in the early stage of network optimization decreases exponentially through iterations. With the increasing of iterations and updates, the long-term speaker centroids will be more reliable and robust. Different from center loss [14], which only pulls the embeddings from the same speaker close to their centers, LSTSL simultaneously enlarges the inter-class differences and reduces the intra-class variations of the learned embeddings. In comparison with Generalized End-to-End Loss (GE2E) [15], which uses mini-batch that consists of a specific number of speakers and utterances, the LSTSL takes the randomly composed mini-batch data as input, which makes the training process more flexible.

4. Experiment and analysis

4.1. Dataset

In our experiments, VoxCeleb [8, 9] dataset is used to investigate the effectiveness of the SV systems, respectively. We adopt the same strategies as [9]. Specifically, the VoxCeleb1 dev and VoxCeleb2 are used as training set. The VoxCeleb1-E is used to evaluate the performance of our system.

4.2. Implementation details

In order to compare experimental results equitably, we decide to make our experimental settings consistent with those of baselines [21], except for the loss functions.

Network structure: The network is modified from the original x-vector [13]. To be specific, a 5-layer TDNN is used to produce frame-level features. Followed [5, 21], the kernel size for each layer is [5,5,7,1,1] without dilation. The output of the last hidden layer is extracted as the segment-level speaker embedding.

Features: The acoustic features are MFCCs with a frame length of 25ms. Mean-normalization is used in each feature dimension of the MFCCs. Moreover, an energy-based voice active detection (VAD) is used to detect speech frames. To increase the diversity of the training data, we augment the training data using reverberation and additive noises from MUSAN [22] and RIR [23], respectively.

Training: Our system is optimized by stochastic gradient descent(SGD), where the initial learning rate is 0.01. L2-regularization is added to prevent overfitting during the training.

Metric: Equal error rate (EER) and minimum detection cost function (minDCF) are used to evaluate the system performance. We use the same value as [8], where the target probability P_{tar} is 0.01, C_{fa} and C_{fr} has the same weight of 1.0.

Table 1: *Results for speaker verification evaluated on VoxCeleb1-E dataset.*

Front-end model	Loss function	EER(%)	minDCF
ResNet50 [9]	contrastive loss	4.19	N/R
R-MSA(3-4) [16]	DALoss-C	4.09	0.458
ResNet-34-SE [17]	AS-softmax	3.10	N/R
ResNet-50 [18]	EAM-softmax	2.94	0.278
ResNet [19]	L_2-softmax	2.38	N/R
x-vector(DDB) [20]	softmax loss	2.31	0.268
x-vector[5]	AMSoftmax+MHE	2.00	N/R
x-vector	Softmax loss	3.28	0.335
x-vector(ours)	LSTSL	**1.98**	**0.189**

Table 2: *Comparison of the proposed LSTSL with different state-of-the-art loss functions on VoxCeleb1-E dataset.*

System	EER(%)
x-vector + Softmax	3.28
x-vector + Asoftmax	2.06
x-vector + AMSoftmax	2.25
x-vector + ArcSoftmax	2.31
x-vector + LMCL	2.19
x-vector + LSTSL($\alpha = 0$)	2.34
x-vector + LSTSL($\alpha = 0.3$)	2.09
x-vector + LSTSL($\alpha = 0.5$)	**1.98**
x-vector + LSTSL($\alpha = 0.7$)	2.04

4.3. Comparison with state-of-the-art loss functions

The experimental results on VoxCeleb1-E are listed in Tables 1. With the same x-vector feature extractor, the x-vector+LSTSL outperforms the x-vecor+softmax by a relative 41.46% EER reduction (1.98 v.s. 3.28). This is mainly because the optimization of LSTSL focuses continuously on the compression of intra-speaker variation and difference of inter-speaker variation, simultaneously. It also outperforms the state-of-the-art AMSoftmax with MHE loss [5]. This suggests that optimized by LSTSL, the x-vector feature extractor can generate more discriminative speaker embeddings.

4.4. Effects of hyperparameter

Besides the softmax loss, we also compare the performance of the proposed LSTSL with the state-of-the-art loss functions. For additive angular margin softmax (Arcsoftmax) [24] and additive margin softmax loss (AMsoftmax) [25], according to the best hyperparameters setting discribed in [5], we set $m = 0.25$ and $m = 0.2$, respectively. For large margin cosine loss (LMSL) [26] the hyperparameters m is set to 0.35. For AM-Softmax, we set $m = 2$. We adopted the cosine similarity as backend for all comparison systems.

The results are shown in Table 2. Compared to the Softmax loss, all other loss functions achieve significant performance gains, which demonstrates the effectiveness of these loss functions in capturing discriminative speaker embeddings. The weight α plays an important role to balance the short-term and long-term update ratio. It is noted that the performance with $\alpha > 0$ is much better than that with $\alpha = 0$. This is because that the short-term speaker embedding centroids only focuses on the discriminating between the speaker centroids within a batch data instead of the all training data. In addition, if there is only one speech segment for each speaker in a batch, the local loss will be reduced to standard affinity loss, which will greatly influence the intra-speaker variance. When α is set to 0.5, we achieve the lowest EER (1.98%) among all the systems. The DET curves of the comparison systems are plotted in Fig. 2. As we can see from the figure, the proposed LSTSL based system exhibits the best performance at most points.

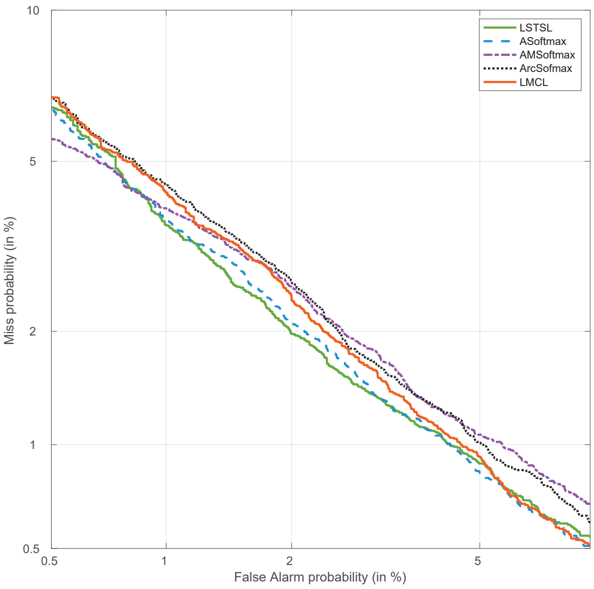

Figure 2: *DET curves for different speaker embedding systems on VoxCeleb1-E.*

5. Conclusions

In this paper, we propose a novel loss function, named long short term speaker loss (LSTSL) for end-to-end speaker verification. The optimization process is based on the comparison between iteratively updated speaker embedding centroids and samples, which is more robust and discriminative than randomly constructed pairs. It is noted that unlike center loss that only pulls the embeddings from the same speaker close to their centers, LSTSL simultaneously enlarges the inter-speaker differences and reduces the intra-speaker variations of the learned embeddings. The experimental results on VoxCeleb dataset demonstrate that the proposed loss function is comparable to the state-of-the-art loss functions.

6. References

[1] J. P. Campbell, "Speaker recognition: A tutorial," *Proceedings of the IEEE*, vol. 85, no. 9, pp. 1437–1462, 1997.

[2] 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.

[3] E. Variani, X. Lei, E. McDermott, I. L. Moreno, and J. 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.

[4] Z. Huang, S. Wang, and K. Yu, "Angular softmax for short-duration text-independent speaker verification." in *Interspeech*, 2018, pp. 3623–3627.

[5] Y. Liu, L. He, and J. Liu, "Large margin softmax loss for speaker verification," *Proc. Interspeech 2019*, pp. 2873–2877, 2019.

[6] J. Peng, Y. Zou, N. Li, D. Tuo, D. Su, M. Yu, C. Zhang, and D. Yu, "Syllable-dependent discriminative learning for small footprint text-dependent speaker verification," in *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2019, pp. 350–357.

[7] R. Li, N. Li, D. Tuo, M. Yu, D. Su, and D. Yu, "Boundary discriminative large margin cosine loss for text-independent speaker verification," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 6321–6325.

[8] A. Nagrani, J. S. Chung, and A. Zisserman, "Voxceleb: A large-scale speaker identification dataset," *Proc. Interspeech 2017*, pp. 2616–2620, 2017.

[9] J. S. Chung, A. Nagrani, and A. Zisserman, "Voxceleb2: Deep speaker recognition," *Proc. Interspeech 2018*, pp. 1086–1090, 2018.

[10] C. Zhang and K. Koishida, "End-to-end text-independent speaker verification with triplet loss on short utterances." in *Interspeech*, 2017, pp. 1487–1491.

[11] S. Novoselov, V. Shchemelinin, A. Shulipa, A. Kozlov, and I. Kremnev, "Triplet loss based cosine similarity metric learning for text-independent speaker recognition." pp. 2242–2246, 2018.

[12] J. Peng, R. Gu, Y. Zou, and W. Wang, "Speaker-discriminative embedding learning via affinity matrix for short utterance speaker verification," in *APSIPA ASC 2019*. IEEE, 2019.

[13] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. 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.

[14] Y. Wen, K. Zhang, Z. Li, and Y. Qiao, "A discriminative feature learning approach for deep face recognition," in *European conference on computer vision*. Springer, 2016, pp. 499–515.

[15] 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.

[16] Z. Gao, Y. Song, I. McLoughlin, P. Li, Y. Jiang, and L. Dai, "Improving aggregation and loss function for better embedding learning in end-to-end speaker verification system," *Proc. Interspeech 2019*, pp. 361–365, 2019.

[17] J. Zhou, T. Jiang, Z. Li, L. Li, and Q. Hong, "Deep speaker embedding extraction with channel-wise feature responses and additive supervision softmax loss function," *Proc. Interspeech 2019*, pp. 2883–2887, 2019.

[18] Y.-Q. Yu, L. Fan, and W.-J. Li, "Ensemble additive margin softmax for speaker verification," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 6046–6050.

[19] T. Zhou, Y. Zhao, J. Li, Y. Gong, and J. Wu, "Cnn with phonetic attention for text-independent speaker verification," in *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2019, pp. 718–725.

[20] Y. Jiang, Y. Song, I. McLoughlin, Z. Gao, and L. Dai, "An effective deep embedding learning architecture for speaker verification," *Proc. Interspeech 2019*, pp. 4040–4044, 2019.

[21] H. Zeinali, L. Burget, J. Rohdin, T. Stafylakis, and J. H. Cernocky, "How to improve your speaker embeddings extractor in generic toolkits," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 6141–6145.

[22] D. Snyder, G. Chen, and D. Povey, "Musan: A music, speech, and noise corpus," *arXiv preprint arXiv:1510.08484*, 2015.

[23] T. Ko, V. Peddinti, D. Povey, M. L. Seltzer, and S. 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.

[24] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "Arcface: Additive angular margin loss for deep face recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4690–4699.

[25] F. Wang, J. Cheng, W. Liu, and H. Liu, "Additive margin softmax for face verification," *IEEE Signal Processing Letters*, vol. 25, no. 7, pp. 926–930, 2018.

[26] 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.