Optimizing Speaker Embeddings using Meta-Training Sets

Nakamasa Inoue, Keita Goto
Tokyo Institute of Technology
E-mail: inoue@c.titech.ac.jp, goto.k.al@m.titech.ac.jp

***Abstract*—This paper presents a method to learn speaker embeddings for text-independent speaker verification. The proposed method aims to optimize embeddings for unseen enrollment/test speakers by training a network with a meta-training set. The main procedure consists of two steps. The first step generates a meta-training set, a set of episodes each with a pair of intra-episode training and testing sets. The second step optimizes network parameters so that the average verification performance over the generated episodes is maximized. An advantage of our approach lies in its complementarity to studies focusing on network structure and we demonstrate its effectiveness with recent ResNet-based models in experiments on the VoxCeleb dataset.**

Index Terms: Text-Independent Speaker Verification, Speaker Embedding, Neural Networks

I. Introduction

Text-independent speaker verification is an important research topic in the field of audio and speech processing with a wide-range of applications such as biometric authentication in web services. It is also known to be a challenging task due to speaker and noise variabilities.

Recently it has been common to approach this task using a statistical methodology aiming to learn speaker embeddings from a large-scale dataset. For example, neural networks trained on the VoxCeleb dataset [1] with more than 1 million utterances can often extract reasonable embeddings for speaker verification at some hidden layers. With this approach, many studies have focused on network structure, and have proven that deep networks including time-delay neural networks (T-DNNs) [2] and residual convolutional networks (ResNets) [3], [4] outperform shallow models including i-vectors with a Gaussian mixture model.

To optimize network parameters, softmax loss or an extension thereof such as AM/AAM softmax loss [5], [6] is widely utilized. This means that, in the training phase, networks are optimized to solve a speaker classification (identification) problem on a given set of utterances with their labels of speaker IDs. This training framework is empirically effective for text-independent speaker verification. However, optimizing embeddings for speaker verification is still recognized as a difficult problem because enrollment/test utterances are assumed to be from new speakers. This is in contrast to the standard classification setting where samples from new categories are out of focus, and also implies that it should be possible to improve the process of training.

On the other hand, some new learning frameworks such as meta-learning [7], [8] and zero-shot learning [9] have focused on effective training for new categories. For example, meta-learning is effective for recognizing new object categories from images [7]. These methods are not always applicable to large-scale speaker verification, because their main application is often to low-resource learning and they are more computationally demanding than the standard learning framework. However, the basic idea to optimize models for new categories by introducing a meta-training set having *episodes* may be effective for optimizing embeddings for unseen speakers for enrollment and testing in speaker verification.

In what follows, we propose a method to optimize speaker embeddings using a meta-training set. Here, a meta-training set is a set of episodes, each of which consists of a pair of intra-episode training and testing sets. More specifically, the proposed method consists of two steps. First, it generates a meta-training set by constructing subsets of utterances from a given training set. Second, it optimizes network parameters so that the average verification performance over the generated episodes is maximized. Note that an advantage of our approach is its complementarity to studies focusing on network structure. In our experiments, we demonstrate the effectiveness of our method using recent ResNet-based models on the VoxCeleb dataset.

The remainder of the paper is organized as follows. Sec. 2 reviews related work on speaker verification. Sec. 3 presents our method with a definition of the meta-training set. Sec. 4 shows experimental results with some discussion. Finally, conclusions are offered in Sec. 5.

II. Related work

Speaker verification is a task to determine whether enrollment and test utterances are from the same speaker. Since our focus is on the text-independent condition, this section reviews models, optimization techniques, and datasets for that condition.

A. Models for Text-Independent Speaker Verification

Over the last ten years, data-driven approaches with probabilistic models have led to great success in text-independent speaker verification. For example, i-vectors [10] extracted from a Gaussian mixture model effectively embed speaker characteristics into fixed-length vectors. This model is based

978-988-14768-8-3/20/$31.00 ©2020 APSIPA

on factor analysis. To further improve the verification performance with i-vectors, probabilistic linear discriminant analysis [11] is often applied.

Various types of neural networks are also proposed to extract speaker embeddings. Snyder et al. [2] report that time delay neural networks (T-DNN) with shift-invariant structures outperform i-vector based systems. Convolutional networks with residual connections, so-called ResNets, have recently been found to be effective with some modifications, e.g., quantization-based aggregation in the VGG model [3], r-vectors extracted from ResNets without max pooling [4], Squeeze-and-Excitation ResNets [12], and a shortcut connected structure for ResNet (SC-ResNet) [13].

B. Optimization Techniques

To extract speaker embeddings from the models described above, model parameters need to be tuned before enrollment utterances proceed to verification systems. Since the number of model parameters is often large, e.g., more than a million for recent networks, statistical learning approaches using a large-scale dataset are effective for estimating parameters.

Assuming that a training dataset consists of utterances with their speaker ID labels, model parameters are typically optimized by solving a speaker identification problem. For the objective function, the softmax (cross-entropy) loss is a standard stable choice to train a network from scratch. A number of studies report angle-based modifications of the softmax loss, for example AM softmax [5], AAM softmax [14], large margin loss [15], and HME/Ring loss [6] are effective for measuring the similarity between two utterances via the cosine similarity of extracted embeddings.

Some recent studies focus on different types of learning frameworks. To improve the robustness against noise, supervised learning with data augmentation is effective. Examples in this respect include augmentation with additional noise datasets [2], augmentation by VAE [16], and mixup learning strategies [17]. To improve cost efficiency, self-supervised learning utilizes unlabeled utterances [18], and multi-task learning shares different tasks [19]. Further, score normalization techniques, such as z-/t-norm [20], adaptive s-norm [21], and cohort score normalization[22] are advantageous for adapting trained models to testing conditions.

C. Datasets

To evaluate verification systems, publicly available datasets are often used. McLaren et al. [23] provided the SITW dataset covering recordings of 299 speakers. Nagrani et al. [1] created the VoxCeleb dataset, which consists of more than 1 million utterances for 5,000+ speakers and is one of the largest datasets for speaker verification and identification. The NIST SRE datasets [24] are leading benchmarks with their workshop series. Since model parameter optimization requires a large number of utterances, VoxCeleb or NIST SRE-08/10 is used for training in practice, and the other small datasets are used for testing.

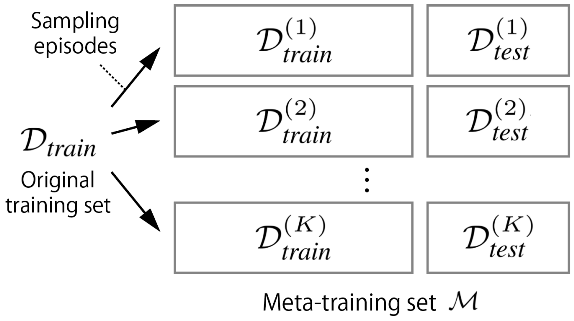
Sampling episodes
$\mathcal{D}_{train}$
Original training set
$\mathcal{D}^{(1)}_{train}$
$\mathcal{D}^{(1)}_{test}$
$\mathcal{D}^{(2)}_{train}$
$\mathcal{D}^{(2)}_{test}$
$\mathcal{D}^{(K)}_{train}$
$\mathcal{D}^{(K)}_{test}$
Meta-training set $\mathcal{M}$

Fig. 1. Generating a meta-training set by sampling episodes.

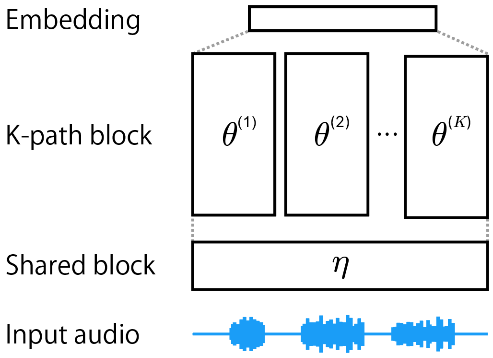

Fig. 2. Meta-network architecture.

III. Proposed Method

Let $\mathcal{D}_{train} = \{(u_i, y_i)\}_{i=1}^{N}$ be a training set consisting of pairs of an utterance u_i and its speaker label y_i. Our goal is to learn embedding function $\phi(u)$ from $\mathcal{D}_{train}$ for text-independent speaker verification. Here, a testing set consists of triplets of an enrollment utterance u_{en}, a test utterance u_{ts}, and their identity ground-truth label g, i.e., $\mathcal{D}_{test} = \{(u_{en,j}, u_{ts,j}, g_j)\}_{j=1}^{M}$ where g_j is 1 if the speakers of $u_{en,j}$ and $u_{ts,j}$ are the same, and otherwise 0. The equal error rate (EER) on $\mathcal{D}_{test}$ is a popular measure for evaluation.

To obtain embedding function $\phi(u)$, a recent trend has involved training a neural network $\mathcal{N}_\theta$, and then extracting features from one of its hidden layers. The network parameter θ is typically optimized by solving a speaker classification problem on $\mathcal{D}_{train}$. However, learning embeddings for speaker verification is still difficult because sets of speakers for training and testing are assumed to be disjoint.

To tackle this difficulty, our proposed method utilizes a meta-training set to optimize embeddings. In the following subsections, we describe 1) the definition of a meta-training set, 2) meta-network architecture, and 3) optimization.

A. Definition of a Meta-Training Set

A meta-training set is defined as a set of *episodes*, each of which simulates a training-testing procedure of speaker verification. Specifically, a meta-training set $\mathcal{M}$ is defined by

$$\mathcal{M} = \{\mathcal{T}_k\}_{k=1}^{K}, \tag{1}$$

where each episode $\mathcal{T}_k$ is a pair of intra-episode training and testing sets given by $\mathcal{T}_k = (\mathcal{D}^{(k)}_{\mathit{train}}, \mathcal{D}^{(k)}_{\mathit{test}})$. All episodes are generated from an episode generator q, which divides the original training set $\mathcal{D}_{\mathit{train}}$ into two subsets for intra-episode training and testing (Figure 1).

Given a meta-training set, our framework seeks a network parameter θ that maximizes the average performance over $\mathcal{D}^{(k)}_{\mathit{test}}$. For example, with EER as an evaluation measure, we choose

$$\theta = \operatorname*{argmin}_{\theta \in \Theta} \sum_{k=1}^{K} \mathrm{EER}(\mathcal{D}^{(k)}_{\mathit{test}}) \tag{2}$$

from a candidate set of parameters Θ (details about how to obtain Θ are given in Sec. III-C). Notably, this framework enables minimization of the expected EER, i.e., $\mathbb{E}_{\mathcal{T} \sim q}\left[\mathrm{EER}(\mathcal{D}_{\mathit{test}})\right]$ as K increases. This means that designing q close to the original training-testing condition helps improve the performance on the original testing set. Thus, we use a two-step episode generator q, which first randomly splits speakers in $\mathcal{D}_{\mathit{train}}$ into two subsets, and then intra-episode training and testing sets are constructed from their corresponding utterances.

B. Meta-Network Architecture

Our framework utilizes a neural network $\mathcal{N}_\theta$, from which K subnetworks $\mathcal{N}^{(k)} (k = 1, 2, \cdots, K)$ are definable. Here, the input-output size of all subnetworks is the same as $\mathcal{N}_\theta$, and each subnetwork corresponds to an episode in a meta-training set in the optimization step.

For simplicity, we use a network architecture with K paths as shown in Figure 2. This architecture consists of two blocks: a shared block and a K-path block. The shared block has a parameter η. This block includes pre-processing and may include low-level feature extraction layers. The K-path block has K independent parameters $\theta^{(1)}, \cdots, \theta^{(K)}$. The embedding layer is put on top of this block. In summary, the network parameter is given by $\theta = (\eta, \theta^{(1)}, \cdots, \theta^{(K)})$, and a subnetwork $\mathcal{N}^{(k)}$ is a network having a parameter $(\eta, \theta^{(k)})$.

Note that this architecture is meta-architecture, and thus recent network structures such as ResNets [3], [4] and SCResNets [13] can be introduced to it. As such, our work is complementary to studies on these network structures.

C. Optimization

The optimization algorithm consists of four steps. First, a network is pre-trained on the entire training set $\mathcal{D}_{\mathit{train}}$. This step is used only for fixing shared parameter η for extracting low-level features. Softmax loss for speaker classification is used in practice. Second, a meta-training set is generated by using the episode distribution q, i.e., generate episodes $\mathcal{T}_k \sim q$ for $k = 1, 2, \cdots, K$, where K is a hyper-parameter of the algorithm. Third, on each intra-episode training set, the corresponding subnetwork $\mathcal{N}_k$ is optimized. Here, $\theta^{(k)}$ is optimized from scratch, and its update history is retained in $\Theta^{(k)}$ at every n iteration. Finally, the parameter θ which minimizes the average EER over intra-episode testing sets is chosen from a candidate set constructed from the histories of parameter updates in the previous step. This optimization process is summarized in Algorithm 1.

Algorithm 1

```
Input: Training set $\mathcal{D}_{\mathit{train}}$
Output: Embedding $\phi(\cdot)$
    Pre-train $\eta$ on $\mathcal{D}_{\mathit{train}}$
    for $k = 1, 2, \cdots, K$ do
        Sample an episode $\mathcal{T}_k = (\mathcal{D}^{(k)}_{\mathit{train}}, \mathcal{D}^{(k)}_{\mathit{test}}) \sim q(\mathcal{T}; \mathcal{D}_{\mathit{train}})$
        $\Theta^{(k)} \leftarrow \mathrm{History}\left[\operatorname*{minimize}_{\theta^{(k)}} \mathrm{Loss}(\theta^{(k)}; \mathcal{D}^{(k)}_{\mathit{train}}, \mathcal{N}^{(k)})\right]$
    end for
    $\Theta \leftarrow \{(\eta, \theta^{(1)}, \cdots, \theta^{(K)}) : \theta^{(k)} \in \Theta^{(k)}\}$
    $\theta \leftarrow \operatorname*{argmin}_{\theta \in \Theta} \sum_{k=1}^{K} \mathrm{EER}(\mathcal{D}^{(k)}_{\mathit{test}}, \bar{\mathcal{N}_\theta})$
    return $\phi(u) := \bar{\mathcal{N}_\theta}(u)$
```

IV. Experiments

Experiments were conducted to explore the effectiveness of the proposed method. We describe the datasets and evaluation settings before moving onto the results.

A. Datasets and Evaluation Settings

The VoxCeleb 1 and 2 datasets [1] are used in all experiments. Following the evaluation protocol, the development set of VoxCeleb 2 consisting of 1,092,009 utterances from 5,994 speakers is used for training and VoxCeleb 1 is used for testing. The testing set has three conditions: O (the original subset with 40 speakers), H (the hard set with 1,190 speakers), and E (the entire set with 1,251 speakers). There are 37,611, 550,894, and 579,818 enrollment-test pairs for conditions O, H, and E, respectively. Two evaluation measures are used: EER and the minimum detection cost (minDCF) from SRE-08/10 with $p_{\text{target}} = 0.01$ and 0.001.

Further, two types of network structure are subjected to evaluation: ResNet-18/34 and SCResNet-18/34. The former network structure is the same as the ResNets for r-vectors [4]. To purely evaluate the effectiveness of meta-traning sets, we did not apply additional techniques (two-step fine-tuning, augmentation with additional datasets, and score normalization) that require hyper-parameter tuning. The latter network structure adds a shortcut connection to the embedding layer as proposed in [13]. The pre-processing layer to extract 40-dim mel-filter bank features with VAD is shared, and one of the aforementioned ResNet-based structures is implemented on each path of the proposed architecture. Each episode uses randomly sampled 200 speakers for testing and the remaining speakers for training. We used Kaldi and TensorFlow to optimize parameters with the momentum SGD optimizer.

TABLE I

PERFORMANCE COMPARISON ON VOXCELEB TEST SETS. THE NETWORK STRUCTURE FOR RESNET18/34 IS FROM [4]. SCRESNET18/34 ADDS THE SHORTCUT CONNECTION PROPOSED IN [13] TO THE EMBEDDING LAYER. MT- DENOTES OUR METHOD USING A META-TRAINING SET WITH $K = 8$. EVALUATION MEASURES ARE THE EQUAL ERROR RATE (EER) AND THE MINIMUM DECISION COST FUNCTION (MINDCF) WITH THE PRIOR TARGET PROBABILITIES $p_1 = 0.01$ AND $p_2 = 0.001$.

	Voxceleb1 Test-O			Voxceleb1 Test-H			Voxceleb1 Test-E		
Method	EER	mDCF$_{p_1}$	mDCF$_{p_2}$	EER	mDCF$_{p_1}$	mDCF$_{p_2}$	EER	mDCF$_{p_1}$	mDCF$_{p_2}$
ResNet18	1.86	0.109	0.290	3.60	0.179	0.554	2.00	0.109	0.412
MT-ResNet18	1.50	0.090	0.245	3.11	0.154	0.488	1.72	0.093	0.351
ResNet34	1.73	0.100	0.270	3.44	0.170	0.525	1.89	0.102	0.395
MT-ResNet34	1.44	0.084	**0.175**	**2.97**	**0.145**	**0.464**	1.64	0.087	**0.341**
SCResNet18	1.76	0.104	0.313	3.52	0.175	0.546	1.91	0.105	0.401
MT-SCResNet18	1.54	0.096	0.242	3.18	0.158	0.500	1.73	0.093	0.359
SCResNet34	1.60	0.093	0.266	3.30	0.164	0.520	1.77	0.097	0.382
MT-SCResNet34	**1.42**	**0.082**	0.185	**2.97**	0.146	0.479	**1.60**	**0.086**	0.343

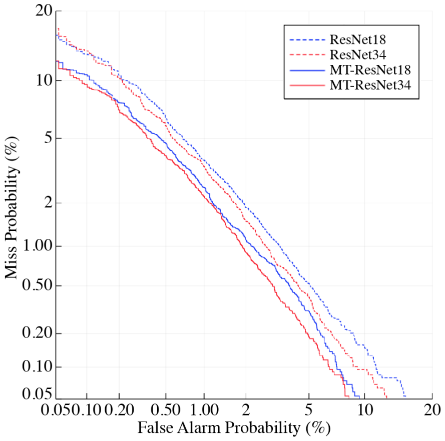

Fig. 3. Detection error tradeoff (DET) curves for ResNet18/34 and MT-ResNet18/34 (proposed method) on VoxCeleb-O.

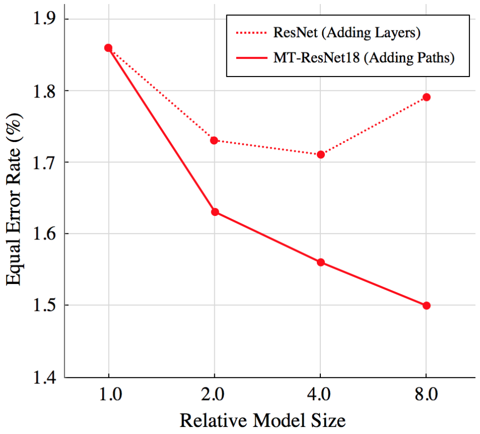

Fig. 4. Tradeoff between verification performance and model size. Two strategies are compared: 1) adding layers to ResNet, 2) adding paths to MT-ResNet18. The equal error rate (%) on VoxCeleb-O is reported. Relative model size is with respect to the size of ResNet18.

B. *Experimental Results*

Table I reports experimental results with and without the proposed method (MT- indicates our method). It can be discerned that the verification performance of all testing conditions is improved with a 9.7%−20% relative reduction in EER and 7.7%−34% reduction in minDCF. The results also confirm that our work is complementary to studies on network structure because performance improvements were observed across all network structures.

For more detailed evaluation, Figure 3 shows detection error tradeoff (DET) curves for ResNet18/34 and MT-ResNet18/34 on the VoxCeleb-O set. As can be seen, the proposed method uniformly improves performance. This means that the proposed method does not overfit to a specific evaluation measure such as EER. We also notice that networks with 34 layers outperform those with 18 layers.

However, a limitation of the proposed method lies in its computational cost. To extract embeddings via K paths in the proposed meta-architecture, a roughly K times greater cost is required because the model size linearly increases. Therefore, we investigate the tradeoff between verification performance and model size in Figure 4. If we compare ResNet-34 and MT-ResNet18 with $K = 2$, having almost the same number of parameters, MT-ResNet18 performs better. This shows that our method using a meta-training set provides a different way to efficiently improve performance, rather than just adding layers.

V. CONCLUSIONS

This paper presented a method to optimize speaker embeddings using a meta-training set for text-independent speaker verification. In experiments, we demonstrated the effectiveness of the proposed method by implementing it on high-performance baselines using ResNets and SCResNets on the VoxCeleb dataset. We also showed that adding paths to the proposed meta-architecture is an efficient way to improve verification performance compared with adding layers to ResNet18.

A useful line of inquiry for future work would be to focus on data-efficient learning frameworks such as semi-supervised and weakly supervised learning as well as data augmentation for low-resource speaker verification.

VI. ACKNOWLEDGEMENT

This work is supported by JST ACT-X Grant No. JPMJAX1905.

REFERENCES

[1] A. Nagrani, J.S. Chung, W. Xie, A. Zisserman, "Voxceleb: Large-scale speaker verification in the wild," Elsevier Computer Speech & Language, vol. 60, pp. 1–15, 2020.

[2] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-Vectors: Robust DNN Embeddings for Speaker Recognition," *Proc. ICASSP*, pp. 5329–5333, 2018.

[3] W. Xie, A. Nagrani, J.S. Chung, and A. Zisserman, "Utterance-level Aggregation for Speaker Recognition in the Wild," *Proc. ICASSP*, pp. 5791–5795, 2019.

[4] H. Zeinali, S. Wang, A. Silnova, P. Matejka, and O. Plchot "BUT System Description to VoxCeleb Speaker Recognition Challenge 2019," CoRR abs/1910.12592, 2019.

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

[6] Y. Liu, L. He, and J. Liu, "Large Margin Softmax Loss for Speaker Verification", *Proc. Interspeech*, pp. 2873–2877, 2019.

[7] S. Ravi, and H. Larochelle, "Optimization as a Model for Few-Shot Learning," *Proc. ICLR*, 2017.

[8] J. Chien, and W.X. Lieow, "Meta Learning for Hyperparameter Optimization in Dialogue System", *Proc. Interspeech*, pp. 839–843, 2019.

[9] K. Williams, "Zero Shot Intent Classification Using Long-Short Term Memory Networks", *Proc. Interspeech*, pp. 844–848, 2019.

[10] N. Dehak, P.J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," IEEE Trans. Audio Speech Lang. Process., vol. 19 (4), pp. 788–798, 2011.

[11] S. Ioffe, "Probabilistic linear discriminant analysis," *Proc. ECCV*, pp. 531–542, 2006.

[12] 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*, pp. 2883–2887, 2019.

[13] S. Seo, D.J. Rim, M. Lim, D. Lee, H. Park, J. Oh, C. Kim, and J. Kim, "Shortcut Connections Based Deep Speaker Embeddings for End-to-End Speaker Verification System", *Proc. Interspeech*, pp. 2928–2932, 2019.

[14] X. Xiang, S. Wang, H. Huang, Y. Qian, and K. Yu, "Margin Matters: Towards More Discriminative Deep Neural Network Embeddings for Speaker Recognition," CoRR abs/1906.07317, 2019.

[15] P. Wang, J. Cui, C. Weng, and D. Yu, "Large Margin Training for Attention Based End-to-End Speech Recognition", *Proc. Interspeech*, pp. 246–250, 2019.

[16] Z. Wu, S. Wang, Y. Qian, and K. Yu, "Data Augmentation Using Variational Autoencoder for Embedding Based Speaker Verification", *Proc. Interspeech*, pp. 1163–1167, 2019.

[17] Y. Zhu, T. Ko, and B. Mak, "Mixup Learning Strategies for Text-Independent Speaker Verification", *Proc. Interspeech*, pp. 4345–4349, 2019.

[18] T. Stafylakis, J. Rohdin, O. Plchot, P. Mizera, and L. Burget, "Self-Supervised Speaker Embeddings", *Proc. Interspeech*, pp. 2863–2867, 2019.

[19] L. You, W. Guo, L. Dai, and J. Du, "Multi-Task Learning with High-Order Statistics for x-Vector Based Text-Independent Speaker Verification", *Proc. Interspeech*, pp. 1158–1162, 2019.

[20] P. Kenny, "Bayesian speaker verification with heavy-tailed priors," *Proc. Odyssey (Keynote Presentation)*, 2010.

[21] P. Matejka, O. Novotny, O. Plchot, L. Burget, M.S. Diez, and J. Cernocky, "Analysis of score normalization in multilingual speaker recognition,", *Proc. Interspeech*, pp. 1567–1571, 2017.

[22] A. Nautsch, J. Patino, A. Treiber, T. Stafylakis, P. Mizera, M. Todisco, T. Schneider, and N. Evans, "Privacy-Preserving Speaker Recognition with Cohort Score Normalisation", *Proc. Interspeech*, pp. 2868–2872, 2019.

[23] M. McLaren, A. Lawson, L. Ferrer, D. Castan, and M. Graciarena, "The Speakers in the Wild (SITW) Speaker Recognition Database," *Proc. Interspeech*, 2016.

[24] S.O. Sadjadi, C. Greenberg, E. Singer, D. Reynolds, L. Mason, and J. Hernandez-Cordero, "The 2018 NIST Speaker Recognition Evaluation", *Proc. Interspeech*, pp. 1483–1487, 2019.

[25] D. Snyder, G. Chen, and D. Povey, "MUSAN: A music, speech, and noise corpus," CoRR abs/1510.08484, 2015.

[26] T. Ko, V. Peddinti, D. Povey, M.L. Seltzer, and S. Khudanpur, "A Study on Data Augmentation of Reverberant Speech for Robust Speech Recognition," *Proc. ICASSP*, 2017.

[27] K. Okabe, T. Koshinaka, and K. Shinoda, "Attentive Statistics Pooling for Deep Speaker Embedding," *Proc. Interspeech*, pp. 2252–2256, 2018.

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