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

Length- and Noise-aware Training Techniques for Short-utterance Speaker Recognition

Wenda Chen[1,*], *Jonathan Huang*[2,*,†], *Tobias Bocklet*[1,3]

[1]Intel Labs
[2]Apple Inc.
[3]Technischen Hochschule Nürnberg

wenda.chen@intel.com, jjhuang@apple.com, tobias.bocklet@intel.com

Abstract

Speaker recognition performance has been greatly improved with the emergence of deep learning. Deep neural networks show the capacity to effectively deal with impacts of noise and reverberation, making them attractive to far-field speaker recognition systems. The x-vector framework is a popular choice for generating speaker embeddings in recent literature due to its robust training mechanism and excellent performance in various test sets. In this paper, we start with early work on including invariant representation learning (IRL) to the loss function and modify the approach with centroid alignment (CA) and length variability cost (LVC) techniques to further improve robustness in noisy, far-field applications. This work mainly focuses on improvements for short-duration test utterances (1-8s). We also present improved results on long-duration tasks. In addition, this work discusses a novel self-attention mechanism. On the VOiCES far-field corpus, the combination of the proposed techniques achieves relative improvements of 7.0% for extremely short and 8.2% for full-duration test utterances on equal error rate (EER) over our baseline system.

Index Terms: speaker recognition, invariant representation learning, centroid alignment, x-vector, far-field

1. Introduction

Speaker recognition system have been popularized in consumer devices such as smart speakers and smartphones. In these use cases, the speech utterance of the user during an interaction with the device can be used to perform voice matching. These use cases are particularly challenging with respect to channel degradation. Furthermore, the interactions tend to be short, making recognition even harder. In this paper, we propose techniques to address these issues. Recently, a successful application of deep neural networks to the domain of speaker recognition helped to improve the accuracy significantly [1]. Here, the network is trained end-to-end via triplet loss. Another state of the art system is x-vector extraction [2]. A frame-wise neural network is used as feature generator followed by a segment-based network using pooled frame-wise data. Training is performed via cross-entropy loss on multiple thousand speakers. For recognition, speaker embeddings are extracted and compared with a similarity measure, e.g., cosine distance. Noise robustness is achieved by PLDA [3] and simulation of noisy training data [4].

Recent research evaluated various extensions of the x-vector approach focusing also on higher noise robustness. The statistical pooling was substituted by attention mechanism [5, 6, 7, 8] in order to learn the weighted pooling in a data-driven manner. Angular softmax [9] and large margin losses [10] showed in combination with noise-augmented training data very good results. [11] presented a scoring approach mimicing the PLDA-scoring by discriminative training. [12, 13] added residual blocks in order to allow deeper networks. [14, 15, 16] motivated loss functions in order to teach the network to learn an implicit cleaning of encodings.

Besides noise, the duration of utterances is a factor of high importance for the accuracy. [17] described the use of gammatone features in combination with i-vector on short utterances, [18] trained deep convolutional networks specifically on short utterances. [1] described the use of inception networks by transforming utterances to fixed length spectrograms and training via triplet loss. [19] used stacked gated recurrent units (GRU) to extract utterance-level features followed by residual convolution neural networks (ResCNNs) trained with speaker identity subspace loss that focuses on transforming utterances from the same speaker to the same subspace. [20] used adversarial learning techniques to learn enhanced embeddings from short and long embedding pairs from the same speaker. [21] argued that pooling and phoneme-aware training is harmful, especially for short-utterance SID and used adversarial training to remove phoneme-variability from the speaker embeddings.

This work focuses on enhanced training techniques for x-vector embeddings with a strong focus on short-duration speech segments in heavy far-field and noisy conditions. We start with our previous work [14] on adding additive margin softmax (AM-softmax) and Invariant Representation Learning (IRL) to an x-vector-based SID system. Due to recent improvements, we substitute the statistical pooling by an attention mechanism. We then modify the idea of IRL and introduce improved training techniques: Centroid Alignment (CA) and Length Variability Cost (LVC). These techniques address the variability in utterances due to channel noises and utterance duration, by making the embedding space less sensitive to such perturbations. CA modifies IRL by enforcing small distances of the training utterance to the average embedding (centroid). LVC tries to keep the distance between a speaker's two randomly-selected utterances with different lengths small. The proposed techniques show improvements on both short and full-length VOiCES test utterances which were collected in noisy and reverberant environments [22]. The rest of the paper is organized as follows: Section 2 provides the intuition behind our training techniques. Section 3 details the modifications we made to the x-vector architecture as well as our attention mechanism. Section 4 describes the model training techniques. Section 5 shows the results and discussions of our experiments. We finish the paper with a short conclusion in Section 6.

[*] Equal Contribution
[†] Work done while at Intel Labs

http://dx.doi.org/10.21437/Interspeech.2020-2872

2. Motivation

The intuition behind our model training techniques can be best illustrated by visualizing the 2-D t-SNE plots of a well-trained speaker embedding (Fig. 1). The embedding was been generated from the baseline system described in Section 3.1. The plots show embeddings extracted from random utterances of 10 speakers in LibriSpeech [23]. The full-length clean speech utterances shows distinct, well-separated clusters; no confusion with other speakers is visible. When noise is added to the full-length utterances [b], the speaker clusters are more blurred. Adding clean speech segments truncated to 2s [c] shows higher confusion than [b]. Combining noise and short duration amplifies the effect [d]. To also quantify the growing deviation, we calculated the average standard deviation of the x-vector embeddings (see line *baseline* in Table 4): noisy embeddings show a higher deviation compared to clean, embeddings generated on short utterances have a higher deviation than noisy ones and noisy short further increases the standard deviation. Based on the simple observation that clean speech is 'better' than noisy, and full-length is 'better' than truncated, we hypothesize that we can use the 'better' samples as an additional training target. This idea is the basis for our auxiliary training objectives: for IRL, we align the noisy speech with its corresponding clean version; for LVC, we align a short duration speech with its full-length version; and for CA, we align the noisy or short-duration speech with the cluster centroid of clean full-length speech for each speaker.

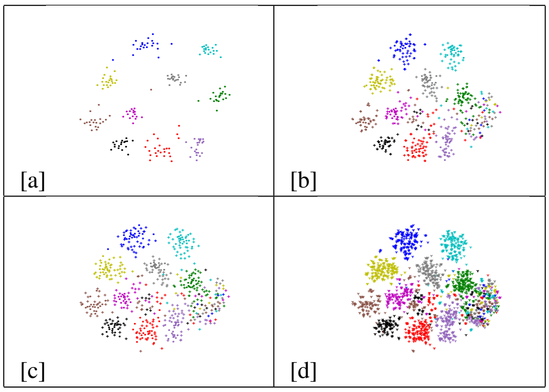

Figure 1: *t-SNE visualization of embeddings for 10 speakers represented by distinct colors. [a]=full-length clean speech; [b]=full-length clean + noisy speech; [c]=full-length clean + short; [d]=all of the above.*

3. System Architectures

3.1. Baseline X-vector Architecture

Although our contributions in attention and model training work for other embedding architectures, we show our findings using the popular x-vector architecture [2]. The topology we used is a slight modification of the original x-vector description. We chose 40-dimension log mel-filterbank as features. The features are mean-normalized on a 3-second sliding window. In Table 1, layers 1-6 are unmodified from the original paper. We removed one of the dense layers before the output, and extract the embedding at layer 7 with 256 dimensions. Our experiments have shown consistently better results with these modifications.

Table 1: *Baseline x-vector system configuration, with N speakers in model training.*

Layer	layer type	layer context	output dimension
1	TDNN-ReLU	[t-2,t+2]	512
2	TDNN-ReLU	{t-2,t,t+2}	512
3	TDNN-ReLU	{t-2,t,t+2}	512
4	Dense-ReLU	{t}	512
5	Dense-ReLU	{t}	1500
6	Stats pool	[0,T]	3000
7	Dense-ReLU	[0,T]	256
8	Dense	[0,T]	N

3.2. Multi-headed Self-attention for X-vector

Attention mechanisms have been proposed in several previous studies [5, 6, 24] to give different weightings to the frames within an speech utterance. In the context of the x-vector embedding, the stats pooling layer is replaced with an attentive pooling layer. The input to the attentive pooling is the hidden representation of layer 5, which we define here as $\mathbf{h}_t$ at time step t. In a departure from previous literature, we compute different attention heads for parts of $\mathbf{h}_t$. Our hypothesis is that the dimensions in this hidden representation correspond to different aspects of the signal which should not be treated with the same weighting. Concretely, we break the vector $\mathbf{h}_t$ into K contiguous, equal-length, smaller sub-vectors $\mathbf{h}_{t,k}$, where $k = 1, ..., K$ is the sub-vector index. The attention weight corresponding to these sub-vectors can be computed by

$$e_{t,k} = f(\mathbf{w}_k^T \mathbf{h}_t + b_k), \tag{1}$$

where $\mathbf{W}_k \in \mathbb{R}^{1500\times1500/K}$ and $b_k \in \mathbb{R}^{1500/K\times1}$ are the trainable weight matrix and bias, respectively, and f is the non-linear activation. We found that the Sigmoid activation here gives the best performance. The frames of the attention weights are normalized by a Softmax

$$\alpha_{t,k} = \frac{exp(e_{t,k})}{\sum_{t=1}^{T} exp(e_{t,k})} \tag{2}$$

The attentive mean and standard deviation pooling for each sub-vector across the entire utterances for $t = 1, ..., T$ time steps is

$$\boldsymbol{\mu}_k = \sum_{t=1}^{K} \alpha_{t,k} \mathbf{h}_{t,k}, \tag{3}$$

and

$$\boldsymbol{\sigma}_k = \sqrt{\sum_{t=1}^{K} \alpha_{t,k} \mathbf{h}_{t,k} \otimes \mathbf{h}_{t,k} - \boldsymbol{\mu}_k \otimes \boldsymbol{\mu}_k}, \tag{4}$$

respectively. $\otimes$ denotes the element-wise multiply operation. Finally, the output of the attentive pooling is formed by a concatenation of the results from 3 and 4, for all k,

$$\mathbf{z} = [\boldsymbol{\mu}_1^T, ..., \boldsymbol{\mu}_K^T, \boldsymbol{\sigma}_1^T, ..., \boldsymbol{\sigma}_K^T]^T \tag{5}$$

as a 3000-dimension vector for the utterance.

4. Training Techniques

4.1. Baseline Training

In the original x-vector paper, the output of the last layer with N speaker labels is fed into a Softmax loss. Because Softmax is specifically designed for classification, others focused on improving the speaker recognition objective with other loss functions [1, 25] or even completely different training infrastructure [26]. In our experiments, the use of Additive Margin Softmax (AM-softmax) loss [10] was consistently superior to basic Softmax training in the far-field test set. The systems trained with AM-softmax did not benefit from having a PLDA backend [3]. We use cosine similarity as the scoring mechanism. This then reflects our baseline system.

For the proposed learning techniques which are describing next, the weights are initialized with the weights of a baseline systems. We used two different methodologies to train our baseline systems: 1. with fixed length 8s long utterances, 2. with utterances of variable length between 0.5 and 8.5s. In the result section, these variants are differentiated by the suffix *-long* for the former and *-varied* for the later variant.

4.2. Invariant Representation Learning

Fig. 2 shows the IRL training procedure. At each training iteration, clean (x) and noisy (x') features of the same utterance are fed into the network one after another, resulting the computation of two AM-softmax loss values. At the layer where the speaker embedding is extracted, we align the two sides by imposing cosine similarity and MSE loss. Mathematically, the combined loss is sumarized as

$$L_{IRL}(x,x') = L_{AM}(x) + \alpha L_{AM}(x') - \gamma L_{cos}(x,x') + \lambda L_2(x,x') \quad (6)$$

where L_{AM} is the categorical loss, L_{cos} is the cosine similarity (7), and L_2 is the means square loss (8)

$$L_{cos}(x,x') = \frac{\phi(x)\cdot\phi(x')}{\|\phi(x)\|\|\phi(x')\|} \quad (7)$$

$$L_2(x,x') = (\phi(x) - \phi(x'))^2 \quad (8)$$

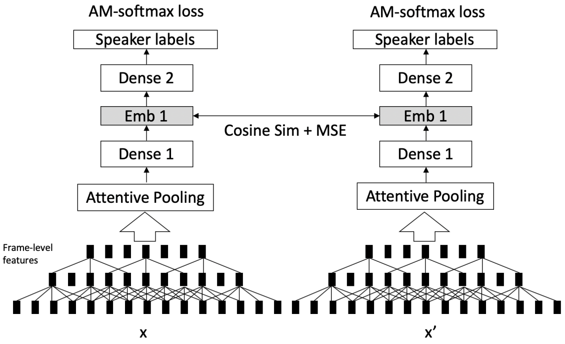

Figure 2: *IRL utilizes parallel clean and noisy speech in each training iteration to train the weights of a single network. The main loss is the AM-softmax, but auxiliary loss functions are introduced to explicitly align the embeddings.*

The embedding layer representation is denoted by ϕ_l. The parameters α, γ, and λ control the contributions from the losses. This training tries to learn a representation invariant to noise. For IRL we achieved best results when initializing with a baseline trained on fixed 8s utterances.

4.3. Length Variability Cost

LVC is in fact a special case of IRL. Here, x is a long-duration utterance (i.e. 8s), and x' is a truncated version of the same utterance (i.e. randomly chosen length between 0.5-8.5s). This training tries to learn a representation invariant to utterance duration. $\alpha = 1$ for LVC and IRL in Eq. 6. For LVC we achieved best results when initializing with a baseline trained by the fixed length methodology.

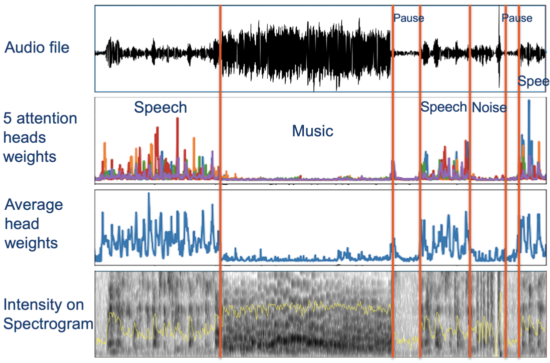

Figure 3: *Attention weights are used as the speech activity detectors.*

4.4. Centroid Alignment

The intuition behind CA is that the clean full-length utterances, and especially their centroid of each speaker, is a desirable training target in the embedding space. After each epoch of training, we compute the embeddings for all clean full-length segments, and average the length-normalized embeddings for each speaker label to form the centroids. For the subsequent epoch, the speaker centroid is the target to do alignment against each training sample. Concretely, for Eq. 6, we compute the AM-softmax for x only, and set α to zero because we do not use training pairs. Instead of doing alignment with another training sample, we use the centroid of speaker for x as the alignment target x'.

5. Experimental Results

5.1. Experimental Settings

Our systems are trained on the Voxceleb 1 and 2 corpora [27]. The training material is prepared by applying 10x data augmentation. For each augmented speech file we convolve a randomly chosen room impulse response (RIR) from 100 artificially generated RIRs by Pyroomacoustics [28] and 100 randomly selected real RIRs from the Aachen Impulse Response Database [29]. Afterwards, the data is mixed in with randomly chosen clips from Google's Audioset under Creative Commons [30].

Table 2: *EER (in %) and minDCF results on VOiCEs Dev and Eval achieved with AM-softmax training on 8s speech segments with and without attention using different x-vector configurations.*

	Dev		Eval	
System	EER	minDCF	EER	minDCF
x-vector	1.78	0.184	5.69	0.374
x-vector-att	1.59	0.190	5.61	0.363
ext-x-vector	1.32	0.149	5.02	0.314
ext-x-vector-att	1.31	0.128	4.99	0.308

Table 3: *EER (in %) and minDCF results on VOiCES Dev for the x-vector-att topology trained with various loss functions(AMSM: AM-softmax, LVC, CA, IRL).* -long *refers to fix 8s segment training;* -varied *refers to training on variable length speech segments (0.5-8.5s). Best result per eval condition are marked in bold.*

Test Duration	AMSM-long		AMSM-varied		LVC		CA		IRL	
1s dev	15.03	0.881	12.71	0.842	13.68	0.857	**11.82**	**0.828**	14.73	0.874
2s dev	7.77	0.658	6.72	0.638	7.13	0.625	**6.59**	**0.618**	7.52	0.650
4s dev	3.81	0.431	4.02	0.457	**3.68**	**0.417**	3.70	0.437	3.83	0.426
8s dev	2.38	0.286	2.50	0.322	**2.09**	**0.270**	2.46	0.308	2.21	0.263.
Full dev	1.59	0.190	1.89	0.233	1.50	0.179	1.78	0.228	**1.46**	**0.171**

The SNR for mixing was uniformly distributed between 0 and 18 dB. We evaluate the proposed loss functions on the VOiCES Dev and Eval data [22] and compare against a baseline system trained with AM-softmax loss. We found optimal weighting factors γ and λ for CA with $\gamma = 0.5$ and $\lambda = 0.01$, meaning the cosine loss is weighted significantly higher. For both LVC and IRL ideal values have been found with $\gamma = \lambda = 0.5$. We see very consistent results between Dev and Eval. For the sake of readability we present numbers on VOiCES dev EER and minDCF ($P_{tar} = 0.01$) metrics.

5.2. Self-attention Model

Fig. 3 visualizes the behaviour of the attention mechanism on an audio segment containing speech, music, and silence. If we look at 5 randomly chosen attention heads, their weights are different in the speech segments, indicating they are learning different aspects of the signal. The average of the attention weights across all heads shows that the silence and noisy segments of the audio are suppressed compared to the speech. Empirically we saw improvements in increasing the number of attention heads to 100, but not beyond. We use 100 heads for the results here. In Table 2 the positive influence of attention is visible in the recognition rates. *x-vector* denotes our baseline system with the topology shown in Table 1. For the sake of completeness, we also show results of *ext-x-vector* with and without attention. *ext-x-vector* is an extended x-vector system as defined in [31]. The temporal context is slightly wider for the frame-level layers and dense layers are added in between the T-DNN layers. The x-vector variants with attention are consistently better than the non-attention variants. The extended variant significantly outperforms the baseline. For the sake of less training time and limited model size and power, we decided to use the smaller topology *x-vector-att* for the evaluation of the proposed loss functions.

5.3. LVC and CA loss

The models are evaluated on various speech durations: 1s, 2s, 4s, 8s, and full (as in VOiCES). Results are summarized in Table 3. For AM-softmax we report results on the training methodologies *-long* (8s training samples) and *-varied* (0.5-8.5s training samples). For AM-softmax, the results on short utterances (1-4s) are superior when trained with variable length methodology. On 8s and full duration, training via long methodology achieved the better results. This confirms the intuition: training a system on long utterances gives better results on longer test utterances. CA shows an advantage in extremely short test utterances (1s and 2s). CA tries to bring short utterances closer to the speaker centroid which is estimated on 8s long utterances and thus helps to stabilize results on short utterances. On 1s test utterances the EER is reduced by 7.0 % compared to AMSM-varied. LVC seems especially helpful for 4s and 8s durations. LVC tries to diminish the distance between long, i.e., 8s and randomly a shorter version (randomly chosen 0.5-8.5s) of the same utterances. Thus, LVC helps to stabilize the training process by removing general speaker variability, which is also visible in Table 4. On 8s segments, LVC improves the EER result by 12.2 % compared to AMSM-long. IRL works best in the full length condition (8s pairs of utterances from the same speaker are aligned) and is close to LVC in the 8s test condition. Trying to tie a clean and noisy utterances of the same size close together, is especially helpful for longer duration utterances: IRL achieves an EER reduction of 8.2 % on the full duration utterances compared to AMSM-long.

Table 4: *Average standard deviation of embeddings for the 10 speakers and conditions described in Fig. 1 before and after CA and LVC techniques.*

	Clean	Noisy	Short	Noisy & short
Baseline	0.0202	0.0284	0.0367	0.0394
After CA	0.0187	0.0282	0.0358	0.0386
After LVC	0.0189	0.0283	0.0349	0.0376

Table 4 shows the average standard deviations of embedding for the 10 speakers plotted in Figure 1 in clean, noisy and short scenarios. Compared to the AM-softmax baseline, CA and LVC help to reduce the variability especially for short and combined (noisy and short) scenarios. For the baseline, the average standard deviations increases for noisy, short and the combination. Applying CA and LVC, brings noisy and short utterances closer to the centroids and removes variation within utterances of one and the same speaker. Hence, making the speaker separation more accurate. While the clean cases also get closer to their cluster centroids, the average relative improvement rate is especially high for the short and noisy cases at 4.6%.

6. Conclusion

In this paper, we proposed to extend the x-vector based SID with self-attention pooling and apply new loss functions based on IRL, CA and LVC. These techniques are targeted for improving speaker recognition on short utterances in reverberant and noisy conditions. Both the proposed attention mechanism and the improved loss functions show a reduction in error rate. Our experiments and analysis proved that the centroids and embedding length variations are good regularization references for the AM-softmax losses. Combining the approaches leads to an EER reduction of 7.0 % on 1s and 8.2 % on full duration utterances.

7. References

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

[2] 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)*, 2018, pp. 5329–5333.

[3] S. Ioffe, "Probabilistic linear discriminant analysis," in *Computer Vision – ECCV 2006*, A. Leonardis, H. Bischof, and A. Pinz, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 531–542.

[4] X. Meng, C. Liu, Z. Zhang, and D. Wang, "Noisy training for deep neural networks," in *2014 IEEE China Summit International Conference on Signal and Information Processing (ChinaSIP)*, 2014, pp. 16–20.

[5] Q. Wang, K. Okabe, K. A. Lee, H. Yamamoto, and T. Koshinaka, "Attention mechanism in speaker recognition: What does it learn in deep speaker embedding?" in *2018 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2018, pp. 1052–1059.

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

[7] K. J. Han, R. Prieto, and T. Ma, "Survey Talk: When Attention Meets Speech Applications: Speech Speaker Recognition Perspective," in *Proc. Interspeech 2019*, 2019.

[8] N. N. An, N. Q. Thanh, and Y. Liu, "Deep cnns with self-attention for speaker identification," *IEEE Access*, vol. 7, pp. 85 327–85 337, 2019.

[9] Z. Huang, S. Wang, and K. Yu, "Angular softmax for short-duration text-independent speaker verification," *Proc. Interspeech, Hyderabad*, 2018.

[10] 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.

[11] L. Ferrer and M. McLaren, "A discriminative condition-aware backend for speaker verification," in *2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 6604–6608.

[12] S. Novoselov, A. Gusev, A. Ivanov, T. Pekhovsky, A. Shulipa, G. Lavrentyeva, V. Volokhov, and A. Kozlov, "STC Speaker Recognition Systems for the VOiCES from a Distance Challenge," in *Proc. Interspeech 2019*, 2019, pp. 2443–2447.

[13] A. Gusev, V. Volokhov, T. Andzhukaev, S. Novoselov, G. Lavrentyeva, M. Volkova, A. Gazizullina, A. Shulipa, A. Gorlanov, A. Avdeeva, A. Ivanov, A. Kozlov, T. Pekhovsky, and Y. Matveev, "Deep speaker embeddings for far-field speaker recognition on short utterances," in *Odyssey 2020 The Speaker and Language Recognition Workshop*, 2020.

[14] J. Huang and T. Bocklet, "Intel Far-Field Speaker Recognition System for VOiCES Challenge 2019," in *Proc. Interspeech 2019*, 2019, pp. 2473–2477.

[15] D. Cai, W. Cai, and M. Li, "Within-sample variability-invariant loss for robust speaker recognition under noisy environments," in *2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 6469–6473.

[16] D. Liang, Z. Huang, and Z. C. Lipton, "Learning noise-invariant representations for robust speech recognition," *IEEE SLT*, 2018.

[17] R. Chakroun and M. Frikha, "Robust features for text-independent speaker recognition with short utterances," *Neural Computing and Applications*, 2020.

[18] G. Bhattacharya, M. J. Alam, and P. Kenny, "Deep speaker embeddings for short-duration speaker verification," in *Proc. Interspeech 2017*, 2017, pp. 1517–1521.

[19] R. Ji, X. Cai, and X. Bo, "An end-to-end text-independent speaker identification system on short utterances," in *Proc. Interspeech 2018*, 2018, pp. 3628–3632.

[20] K. Liu and H. Zhou, "Text-independent speaker verification with adversarial learning on short utterances," in *2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 6569–6573.

[21] N. Tawara, A. Ogawa, T. Iwata, M. Delcroix, and T. Ogawa, "Frame-level phoneme-invariant speaker embedding for text-independent speaker recognition on extremely short utterances," in *2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 6799–6803.

[22] M. K. Nandwana, J. Van Hout, M. McLaren, C. Richey, A. Lawson, and M. A. Barrios, "The VOiCES from a Distance Challenge 2019 Evaluation Plan," *Special Session for Interspeech 2019*, 2019.

[23] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An asr corpus based on public domain audio books," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 5206–5210.

[24] F. R. rahman Chowdhury, Q. Wang, I. L. Moreno, and L. Wan, "Attention-based models for text-dependent speaker verification," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5359–5363.

[25] 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.

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

[27] A. Nagrani, J. S. Chung, W. Xie, and A. Zisserman, "Voxceleb: Large-scale speaker verification in the wild," *Comput. Speech Lang.*, vol. 60, 2020.

[28] R. Scheibler, E. Bezzam, and I. Dokmanić, "Pyroomacoustics: A python package for audio room simulation and array processing algorithms," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 351–355.

[29] M. Jeub, M. Schafer, and P. Vary, "A binaural room impulse response database for the evaluation of dereverberation algorithms," in *2009 16th International Conference on Digital Signal Processing*. IEEE, 2009, pp. 1–5.

[30] J. F. Gemmeke, D. P. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, "Audio set: An ontology and human-labeled dataset for audio events," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 776–780.

[31] D. Snyder, J. Villalba, N. Chen, D. Povey, G. Sell, N. Dehak, and S. Khudanpur, "The JHU Speaker Recognition System for the VOiCES 2019 Challenge," in *Proc. Interspeech 2019*, 2019, pp. 2468–2472.