Phonetic-Attention Scoring for Deep Speaker Features in Speaker Verification

Lantian Li†, Zhiyuan Tang†, Ying Shi†, Dong Wang†*
† Center for Speech and Language Technologies, Tsinghua University, Beijing, China
Corresponding Author E-mail: wangdong99@mails.tsinghua.edu.cn

***Abstract*—Recent studies have shown that frame-level deep speaker features can be derived from a deep neural network with the training target set to discriminate speakers by a short speech segment. By pooling the frame-level features, utterance-level representations, called d-vectors, can be derived and used in the automatic speaker verification (ASV) task. This simple average pooling, however, is inherently sensitive to the phonetic content of the utterance. An interesting idea borrowed from machine translation is the attention-based mechanism, where the contribution of an input word to the translation at a particular time is weighted by an attention score. This score reflects the relevance of the input word and the present translation. We can use the same idea to align utterances with different phonetic contents.**

This paper proposes a phonetic-attention scoring approach for d-vector systems. By this approach, an attention score is computed for each frame pair. This score reflects the similarity of the two frames in phonetic content, and is used to weigh the contribution of this frame pair in the utterance-based scoring. This new scoring approach emphasizes the frame pairs with similar phonetic contents, which essentially provides a soft alignment for utterances with any phonetic contents. Experimental results show that compared with the naive average pooling, this phonetic-attention scoring approach can deliver consistent performance improvement in ASV tasks of both text-dependent and text-independent.

I. Introduction

Automatic speaker verification (ASV) is an important biometric authentication technology and has a broad range of applications. The current ASV approach can be categorized into two groups: the statistical model approach and the neural model approach. The most famous statistical models for ASV involve the Gaussian mixture model-universal background model (GMM-UBM) [1], the joint factor analysis model [2] and the i-vector model [3], [4], [5]. As for the neural model approach, Ehsan et al. proposed the first successful implementation [6], where frame-level speaker features were extracted from a deep neural network (DNN), and utterance-level speaker representations ('d-vectors') were derived by averaging the frame-level features, i.e., average pooling. This work was followed by a bunch of researchers [7], [8], [9], [10].

The neural-based approach is essentially a feature learning approach, i.e., learning frame-level speaker features from raw speech. In previous work, we found that by this feature learning, speakers can be discriminated by a speech segment as short as 0.3 seconds [10], either a word or a cough [11]. However, with the conventional d-vector pipeline, this brilliant frame-level discriminatory power cannot be fully utilized by the utterance-level ASV, due to the simple average pooling. This shortage was quickly identified by researchers, and hence almost all the studies after Ehsan et al. [6] chose to learn representations of segments rather than frames, the so-called *end-to-end* approach [8], [12], [13], [14]. However, frame-level feature learning possesses its own advantages in both generalizability and ease of training [15], and meets our long-term desire of deciphering speech signals [16]. An ideal approach, therefore, is to keep the feature learning framework but solve the problem caused by average pooling.

To understand the problem of average pooling, first notice that feature pooling is equivalent to score pooling. To make the presentation clear, we consider the simple inner product score:

$$\vec{s}_u \cdot \vec{s}_{u'} = \frac{1}{|u|} \sum_{f \in u} \vec{v}_f \cdot \frac{1}{|u'|} \sum_{f' \in u'} \vec{v}_{f'},$$

where u and u' are two utterances in test, f denotes frames; $\vec{v}_f$ and $\vec{s}_u$ are frame-level speaker features and utterance-level d-vectors, respectively. A simple arrangement leads to:

$$\vec{s}_u \cdot \vec{s}_{u'} = \frac{1}{|u|} \frac{1}{|u'|} \sum_{f \in u} \sum_{f' \in u'} \vec{v}_f \cdot \vec{v}_{f'}.$$

This formula indicates that with average pooling, the utterance-level score $\vec{s}_u \cdot \vec{s}_{u'}$ is the average of the frame-level scores $\vec{v}_f \cdot \vec{v}_{f'}$. Most importantly, the scores of all the frame pairs (f, f') are equally weighted, which is obviously suboptimal, as the reliability of scores from different frame pairs may be substantially different. In particular, a pair of frames in the same phonetic context may result in a much more reliable frame-level score compared to a pair in different phonetic context, as demonstrated by the fact that text-dependent ASV generally outperforms text-independent ASV. This indicates that a key problem of the average pooling method is that phonetic variation may cause serious performance degradation. This partly explains why d-vector systems are mostly successful in text-dependent tasks.

A simple idea is to discriminate frame pairs in similar / different phonetic contents, and put more emphasis on the frame pairs in similar phones. This can be formulated by:

$$\vec{s}_u \cdot \vec{s}_{u'} = \frac{1}{|u|} \frac{1}{|u'|} \sum_{f \in u} \sum_{f' \in u'} \alpha(f, f') \cdot \vec{v}_f \cdot \vec{v}_{f'}, \qquad (1)$$

where $\alpha(f, f')$ represents the weight for the frame pair (f, f'), computed from the similarity of their phonetic contents. This is essentially a soft-alignment approach that aligns two utterances with respect to phonetic contents, where $\alpha(f, f')$ represents the alignment degree of frames f and f', derived from the phonetic information of the two frames.

The idea of soft-alignment was motivated by the *attention mechanism* in neural machine translation (NMT) [17], where the contribution of an input word to the translation at a particular time is weighted by an attention score, and this attention score reflects the relevance of the input word and the present translation. We therefore name our new scoring model by Eq. (1) as *phonetic-attention scoring*. By paying more attention to frame pairs in similar phonetic contents, this new scoring approach essentially turns a text-independent task to a text-dependent task, hence partly solving the problem caused by phone variation with the naive average pooling.

In the next section, we will briefly describe the attention mechanism. The phonetic-attention scoring approach will be presented in Section III, and the experiments will be reported in Section V. The entire paper will be concluded in Section VI.

II. Attention Mechanism

The attention mechanism was firstly proposed by [17] in the framework of sequence to sequence learning, and was applied to NMT. Recently, this model has been widely used in many sequential learning tasks, e.g., speech recognition [18]. In a nutshell, the attention approach looks up all the input elements (e.g., words in a sentence or frames in an utterance) at each decoding time, and computes an attention weight for each element that reflects the relevance of that element with the present decoding. Based on these attention weights, the information of the input elements is collected and used to guide decoding. As shown in Fig. 1, at decoding time t, the attention weight $\alpha_{t,i}$ is computed for each input element $\vec{x}_i$ (more precisely, the annotation of $\vec{x}_i$, denoted by $\vec{h}_i$), formally written as:

$$\alpha_{t,i} = \sigma(g(\vec{z}_{t-1}, \vec{h}_i))$$

where $\vec{z}_{t-1}$ is the decoding status at time t, and g is a *value function* that can be in any form. σ is a normalization function (usually softmax) that ensures $\sum_i \alpha_{t,i} = 1$. The decoding for $\vec{y}_t$ is then formally written as:

$$\vec{y}_t = g'(\vec{z}_{t-1}, \vec{y}_{t-1}, \sum_i \alpha_{t,i}\vec{h}_i),$$

where g' is the decoding model. In the conventional setting, g is a parametric function, e.g., a neural net, whose parameters are jointly optimized with other parts of the model, e.g., the decoding model g'.

III. Phonetic-attention scoring

We borrow the architecture shown in Fig. 1 to build our phonetic-attention model in Eq. (1). Since our purpose is to align two existing sequences rather than sequence to sequence generation, the structure can be largely simplified. For example, the recurrent connection in both the input and output sequence can be omitted. Secondly, in Fig. 1, the value function g is learned from data; for our scoring model, we have a clear goal to align utterances by phonetic content, so we can design the value function by hand (although function learning with prior may help). This leads to the phonetic-attention model shown in Fig. 2.

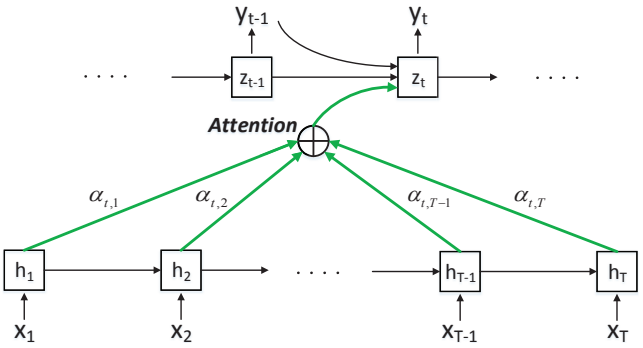

Fig. 1. Attention mechanism in sequence to sequence model.

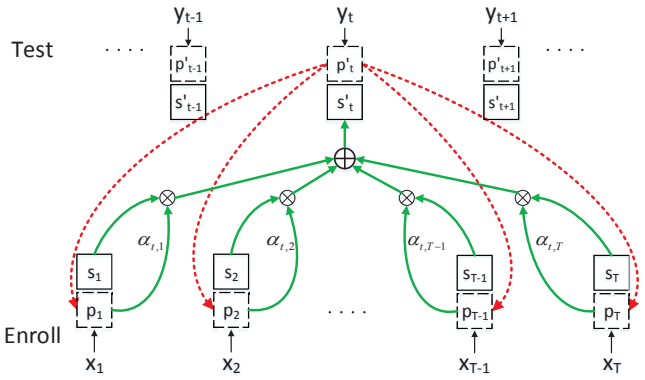

Fig. 2. Diagram of the phonetic-attention model.

The architecture and the associated scoring method can be summarized into the following four steps:

(1) For both the enrollment and test utterances, compute the frame-level speaker features from a speaker recognition DNN, denoted by $S = [\vec{s}_1, \vec{s}_2, ..., \vec{s}_T]$ and $S' = [\vec{s}'_1, \vec{s}'_2, ..., \vec{s}'_{T'}]$. Additionally, compute the frame-level phonetic features from a speech recognition DNN, denoted by $P = [\vec{p}_1, \vec{p}_2, ..., \vec{p}_T]$ and $P' = [\vec{p}'_1, \vec{p}'_2, ..., \vec{p}'_{T'}]$.

(2) For each frame t in the test utterance, compute the attention weight $\alpha_{t,i}$ for each frame i in the enrollment utterance.

$$\alpha_{t,i} = \frac{KL^{-1}(p'_t, p_i)}{\sum_i KL^{-1}(p'_t, p_i)},$$

where the $KL^{-1}(\cdot,\cdot)$ denotes the reciprocal of KL distance. This step is represented by the red dashed line in Fig. 2.

(3) Compute the matching score of frame t in the test utterance as follows:

$$d_t = \sum_i \alpha_{t,i} \cdot cos(s'_t, s_i).$$

This step is represented by the green solid line in Fig. 2.
(4) Compute the matching score of the two utterances by averaging the frame-level matching score:

$$d = \frac{1}{T}\sum_t d_t = \frac{1}{T}\sum_t \sum_i \alpha_{t,i} \cdot cos(s'_t, s_i).$$

IV. Related work

The attention mechanism has been studied by several authors in ASV, e.g., [13], [19], [20], [21], [22], [23]. However, most of the proposals used the attention mechanism to produce a better frame pooling, while we use it to produce a better utterance alignment. In essence, these methods learn which frame should contribute to the speaker embedding, while our approach learn which frame-pair should contribute to the matching score. Moreover, most of these studies do not use phonetic knowledge explicitly, except [13].

Another work relevant to ours is the segmental dynamic time warping (SDTW) approach proposed by Mohamed et al. [24]. This work holds the same idea as ours in aligning frame-level speaker features, however their alignment is based on local temporal continuity, while ours is based on global phonetic contents.

V. Experiments

A. Data

1) Training data: The data used to train the d-vector systems is the *CSLT-7500* database, which was collected by CSLT@Tsinghua University. It consists of $7,500$ speakers and $1,532,766$ utterances. The sampling rate is 16 kHz and the precision is 16-bit. Data augmentation is applied to cover more acoustic conditions, for which the MUSAN corpus [25] is used to provide additive noise, and the room impulse responses (RIRS) corpus [26] is used to generate reverberated samples.

2) Evaluation data: (a) *CIIH*: a dataset contains short commands used in the intelligent home scenario. It contains recordings of 10 short commands from 100 speakers, and each command consists of $2\sim5$ Chinese characters. For each speaker, every command is recorded 15 times, amounting to 150 utterances per speaker. This dataset is used to evaluate the text-dependent (TD) task.
(b) *DSDB*: a dataset involving digital strings. It contains $1,099$ speakers, each speaking $15\sim20$ Chinese digital strings. Each string contains 8 Chinese digits, and is about $2\sim3$ seconds. For each speaker, 5 utterances are randomly sampled as enrollment, and the rest are used for test. This dataset is used to evaluate the text-prompted (TP) task.
(c) *ALI-WILD*: a dataset collected by the Ali crowdsource platform. It covers unlimited real-world scenarios, and contains 669 speakers and $27,861$ speech segments. We designed two test conditions: a short-duration scenario Ali(S) where the duration of the enrollment is 15 seconds and the test is 3 seconds, and a long-duration scenario Ali(L) where the duration of the enrollment is 30 seconds and the test is 15 seconds. This dataset is used to evaluate the text-independent (TI) task.

B. Settings

The DNN model to produce frame-level speaker features is a 9-layer time-delay neural network (TDNN), where the slicing parameters are $\{t\text{-}2, t\text{-}1, t, t\text{+}1, t\text{+}2\}$, $\{t\text{-}2, t\text{+}2\}$, $\{t\}$, $\{t\text{-}1, t\text{+}1\}$, $\{t\}$, $\{t\text{-}2, t\text{+}2\}$, $\{t\}$, $\{t\text{-}4, t\text{+}4\}$, $\{t\}$. Except the last hidden layer that involves 400 neurons, the size of all other layers is $1,000$. Once the DNN has been fully trained, 400-dimensional deep speaker features were extracted from the last hidden layer. The model was trained using the Kaldi toolkit [27]. Based on this model, we built a standard d-vector system with the naive average pooling, denoted by *Baseline*.

The phonetic-attention model requires frame-level phonetic features. We built a DNN-HMM hybrid system using Kaldi following the WSJ S5 recipe. The training used 500 hours of Chinese speech data. The model is a TDNN, and each layer contains 512 nodes. The output layer contains 463 units, corresponding to the number of GMM senones. Once the model was trained, 463-dimensional phone posteriors were derived from the output layer and were used as phonetic features. The phonetic-attention system based on the phone posteriors is denoted by *Att-Post*. Another type of phonetic features can be derived from the final affine layer. To compress the size of the feature vector, the Singular Value Decomposition (SVD) was applied to decompose the final affine matrix into two low-rank matrices, where the rank was set to 100. The 100-dimensional activations were read from the low-rank layer of the decomposed matrix, which we call bottleneck features. The phonetic-attention system based on the bottleneck features is denoted by *Att-BN*.

Finally, we built a phone-blind attention system where the attention weight is computed from the speaker feature itself, rather than phonetic features. This approach is similar to the work in [19], [20], though the attention function is not trained. This system is denoted by *Att-Spk*.

C. Results

The results in terms of the equal error rate (EER) are shown in Table I, where the baseline system is based on the naive average pooling, while the three attention-based systems use attention models based on different features. For each system, it reports results with two frame-level metrics: cosine distance and cosine distance after LDA. The LDA model was trained on CSLT-7500, and the dimensionality of its projection space was set to 150. There are four tasks in total: the TD task on CIIH, the TP task on DSDB, the TI short-duration task on Ali(S), and the TI long-duration task on Ali(L). The best performance is marked in bold face.

From these results, it can be seen that on all these tasks, the attention-based systems outperform the baseline system, indicating that the naive average pooling is indeed problematic. When comparing these three attention-based systems, we find they perform quite different on different tasks. On the TD task

CIIH and TP task DSDB, the phone-blind attention system Att-Spk seems slightly superior, while on the TI task Ali(S) and Ali(L), the two phonetic-attention systems are clearly better. This observation is understandable, as on the TD or the TP tasks, the phonetic variation in enrollment and test utterances are largely identical, so the appropriate alignment can be easily found by even a phone-blind attention. On the TI tasks, however, the phonetic variation is much more complex, for which additional phonetic information is required to align the enrollment and test utterances. Finally, comparing the two phonetic-attention systems, the Att-BN is consistently better. This indicates that the bottleneck feature is a more compact representation for the phonetic content.

TABLE I
PERFORMANCE OF DIFFERENT SYSTEMS ON DIFFERENT TASKS.

Systems	Metric	EER(%)			
		CIIH	DSDB	Ali(S)	Ali(L)
Baseline	Cosine	3.71	1.02	9.24	4.95
	LDA	2.49	0.70	5.84	2.44
Att-Spk	Cosine	3.27	0.95	9.07	4.95
	LDA	**2.11**	**0.65**	5.80	2.50
Att-Post	Cosine	3.28	0.97	9.12	4.85
	LDA	2.22	0.69	5.76	2.32
Att-BN	Cosine	3.20	0.98	9.11	4.84
	LDA	2.18	0.70	**5.69**	**2.31**

D. Analysis

To better understand the difference behavior of the phone-blind attention and the phonetic attention, we draw the alignment produced by them on two samples from the TD and TI tasks respectively. The figures are shown in Fig. 3 and Fig. 4.[1] It can be seen that on the TD task, two attention approaches produce similar alignments, while the alignment produced by phonetic attention is more concentrated. This is not surprising, as the phonetic features are short-term and change more quickly than the speaker features. Actually, this might be a key problem of the present implementation of the phonetic attention, as the concentration means less frames in one utterance being aligned for each frame in the other utterance, leading to unreliable scores. Nevertheless, the explicit phonetic information does provide much more accurate alignments in the TI scenario, where the phonetic variation is complex and phone-blind attention may produce rather poor alignments. This can be seen from Fig. 4 that the aligned segments produced by the phonetic attention show clear slopped patterns, which is more realistic than the flat patterns produced by the phone-blind attention.

VI. CONCLUSIONS

This paper proposed a phonetic-attention scoring approach for the d-vector speaker recognition system. This approach uses frame-level phonetic information to produce a soft alignment between the enrollment and test utterances, and computes the matching score by emphasizing the aligned frame pairs.

[1]The observations of the TD and TP tasks are quite similar, so here the figure on the TP task is omitted.

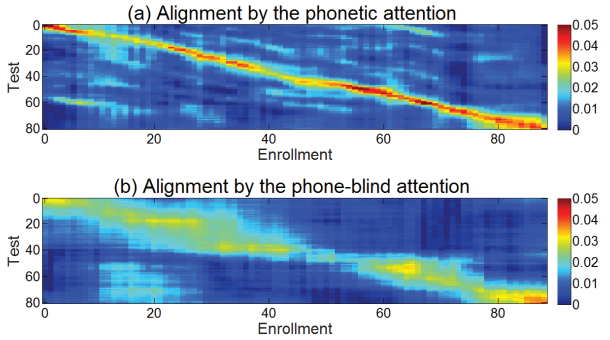

Fig. 3. Alignment produced by the phone-blind and phonetic attentions on the TD task.

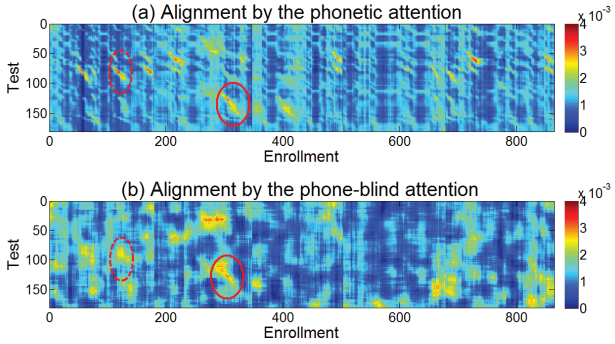

Fig. 4. Alignment produced by the phone-blind and phonetic attentions on the TI task.

We tested the method on text-dependent, text-prompted and text-independent tasks, and found that it delivered consistent performance improvement over the baseline system. The phonetic attention was also compared with a naive phone-blind attention, and the results showed that the phone-blind attention worked well in text-dependent and text-prompt tasks, but failed in text-independent tasks. Analysis was conducted to explain the observation. In the further work, we will study speaker features that change more slowly. e.g., vowel-only feature. It is also interesting to learn the value function.

ACKNOWLEDGEMENT

This work was supported by the National Natural Science Foundation of China No. 61633013, and the Postdoctoral Science Foundation of China No. 2018M640133.

REFERENCES

[1] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker verification using adapted Gaussian mixture models," *Digital signal processing*, vol. 10, no. 1-3, pp. 19–41, 2000.

[2] P. Kenny, G. Boulianne, P. Ouellet, and P. Dumouchel, "Joint factor analysis versus eigenchannels in speaker recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 4, pp. 1435–1447, 2007.

[3] 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.

[4] S. Ioffe, "Probabilistic linear discriminant analysis," *Computer Vision–ECCV*, pp. 531–542, 2006.

[5] Y. Lei, N. Scheffer, L. Ferrer, and M. McLaren, "A novel scheme for speaker recognition using a phonetically-aware deep neural network," in *ICASSP*. IEEE, 2014, pp. 1695–1699.
[6] E. Variani, X. Lei, E. McDermott, I. L. Moreno, and J. Gonzalez-Dominguez, "Deep neural networks for small footprint text-dependent speaker verification," in *ICASSP*. IEEE, 2014, pp. 4052–4056.
[7] Y. Liu, Y. Qian, N. Chen, T. Fu, Y. Zhang, and K. Yu, "Deep feature for text-dependent speaker verification," *Speech Communication*, vol. 73, pp. 1–13, 2015.
[8] 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*. IEEE, 2016, pp. 165–170.
[9] D. Snyder, D. Garcia-Romero, D. Povey, and S. Khudanpur, "Deep neural network embeddings for text-independent speaker verification," in *Interspeech*, 2017, pp. 999–1003.
[10] L. Li, Y. Chen, Y. Shi, Z. Tang, and D. Wang, "Deep speaker feature learning for text-independent speaker verification," in *Interspeech*, 2017, pp. 1542–1546.
[11] M. Zhang, Y. Chen, L. Li, and D. Wang, "Speaker recognition with cough, laugh and" wei"," *arXiv preprint arXiv:1706.07860*, 2017.
[12] G. Heigold, I. Moreno, S. Bengio, and N. Shazeer, "End-to-end text-dependent speaker verification," in *ICASSP*. IEEE, 2016, pp. 5115–5119.
[13] S.-X. Zhang, Z. Chen, Y. Zhao, J. Li, and Y. Gong, "End-to-end attention based text-dependent speaker verification," in *Spoken Language Technology Workshop*. IEEE, 2016, pp. 171–178.
[14] C. Zhang and K. Koishida, "End-to-end text-independent speaker verification with triplet loss on short utterances," in *Interspeech*, 2017.
[15] D. Wang, L. Li, Z. Tang, and T. F. Zheng, "Deep speaker verification: Do we need end to end?" *arXiv preprint arXiv:1706.07859*, 2017.
[16] L. Li, D. Wang, Y. Chen, Y. Shi, Z. Tang, and T. F. Zheng, "Deep factorization for speech signal," in *ICASSP*. IEEE, 2018.
[17] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *ICLR*, 2015.
[18] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *ICASSP*. IEEE, 2016, pp. 4960–4964.
[19] F. R. rahman Chowdhury, Q. Wang, I. L. Moreno, and L. Wan, "Attention-based models for text-dependent speaker verification," in *ICASSP*. IEEE, 2018, pp. 5359–5363.
[20] Y. Liu, L. He, W. Liu, and J. Liu, "Exploring a unified attention-based pooling framework for speaker verification," *arXiv preprint arXiv:1808.07120*, 2018.
[21] Y. Zhu, T. Ko, D. Snyder, B. Mak, and D. Povey, "Self-attentive speaker embeddings for text-independent speaker verification," *Interspeech*, pp. 3573–3577, 2018.
[22] K. Okabe, T. Koshinaka, and K. Shinoda, "Attentive statistics pooling for deep speaker embedding," *arXiv preprint arXiv:1803.10963*, 2018.
[23] Y. Liu, L. He, J. Liu, and M. T. Johnson, "Speaker embedding extraction with phonetic information," *arXiv preprint arXiv:1804.04862*, 2018.
[24] M. Adel, M. Afify, and A. Gaballah, "Text-independent speaker verification based on deep neural networks and segmental dynamic time warping," *arXiv preprint arXiv:1806.09932*, 2018.
[25] D. Snyder, G. Chen, and D. Povey, "MUSAN: A Music, Speech, and Noise Corpus," 2015, arXiv:1510.08484v1.
[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," in *ICASSP*. IEEE, 2017, pp. 5220–5224.
[27] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz *et al.*, "The kaldi speech recognition toolkit," in *Workshop on automatic speech recognition and understanding*. IEEE Signal Processing Society, 2011.