Context-adaptive Gaussian Attention for Text-independent Speaker Verification

Junyi Peng* Rongzhi Gu* Haoran Zhang*and Yuexian Zou*†
* Peking University Shenzhen Graduate School, Shenzhen, China
† Peng Cheng Laboratory, Shenzhen, China
E-mail: zouyx@pku.edu.cn

***Abstract*—Multi-head attention (MHA) has shown its effectiveness on aggregating frame-level features for speaker verification task. However, MHA weights each frame individually without considering context information which is important for modeling speaker characteristics of the speech. Based on the assumption that the highly relevant context information should follow a temporal Gaussian distribution, we propose a novel variant of multi-head attention, named as context-adaptive Gaussian attention (CGA), which employs a set of Gaussian functions with different parameters to dynamically model the distributions of the weights obtained from each head. Furthermore, a Gaussian Clustering algorithm (GC) is designed to merge the overlapped Gaussian distributions between different heads. In this way, the proposed method can facilitate the model to better capture multi-span context information compared to the traditional multi-head attention. Experiments on Voxceleb1 dataset demonstrate that the proposed CGA outperforms the state-of-the-art pooling approaches.**

I. Introduction

Speaker verification (SV) is the task of verifying whether an unkown speech segment belongs to a specific target speaker. According to the restriction of the uttered content, speaker verification can be categorized into text-dependent speaker verification (TD-SV) and text-independent speaker verification (TI-SV).

For many years, the combination of i-vector and Probabilistic Linear Discriminant Analysis (PLDA) has become the dominant approach [1]. Recently, with the advancement in deep learning, more attention has been paid to discriminative speaker embedding learning in speaker verification (SV) task. Among these attempts, well-designed neural networks (e.g. convolutional neural networks (CNNs) [2], recurrent neural networks (RNNs) [3]) and loss function (e.g. triplet loss[4], Angular Softmax loss [5], cosine loss [6], affinity loss[7]) have been employed to enhance the discrimination of speaker embedding. Most of these SV systems employed a pooling mechanism to aggregation the variable-length frame-level features into an utterance-level speaker embedding representation.

In d-vector based systems [8][9][10], temporal average pooling was utilized to average the activation vectors of the bottleneck layer over the feature sequence of an input speech segment. The works in [11][12] proposed to use higher-order statistic to characterize the variation across the frame-level features. The mean and standard deviation of the frame-level feature vectors were computed and then concatenated via a statistic pooling layer. The speaker embedding (referred to as x-vector) was derived from the following two hidden layers. The experiments showed that x-vector outperformed i-vector for short duration speech segments. However, the statistic pooling assigns equal weight to each frame-level feature, this may limit the performance of x-vectors.

Intuitively, the speaker model should be built upon the frames corresponding to the phonemes instead of silent and noisy frames. Thus, many efforts have been focused on using a structured attention layer to learn different weights for different frames. In [13][14], the frame-level weights are learned by a self-attention pooling mechanism, the concatenation of weighted mean and standard deviation vector over an utterance was produced by a weighted statistics pooling layer. To help the network to attend to different sub-sets of the encoded frame-level features, a self multi-head attention (MHA) mechanism was introduced in [3][15]. The MHA produces a set of weight alignments (one for each head), so that the SV model can jointly capture crucial discriminative information from different representation subspaces. In this way, the final utterance-level speaker embedding is obtained by concatenating the utterance-level representations from all the heads. The existing attention mechanisms weight each frame-level feature independently without considering of the context information. However, the impact of context information among frames is important in speech processing literature.

To address this problem, this paper proposes a novel strategy to strengthen multi-head attention through capturing the context information. The key idea is to learn context-adaptive attention for aggregating the frame-level features. Specifically, the weights obtained from each attention head are calibrated using a Gaussian distribution, which is used to dynamically capture the highly relevant regions in a fixed-length context. To make the model attend to context with vairous lengths, a Gaussian Clustering algorithm is further proposed to merge the overlapped Gaussian distributions from different heads. This enables the model to capture multi-span context information. Finally, the revised weights are used to compute the weighted mean vectors and weighted standard deviation (std) vectors over variable-length frame-level features. These weighted mean and std vectors are then concatenated and fed into the following proceeding layers to produce the utterance-level speaker embedding.

In summary, our contributions include: (a) we propose a novel context-adaptive Gaussian attention (CGA) mechanism to weight frame-level features by the context information. (b) a Gaussian Clustering (GC) algorithm is proposed to merge overlapping weights. (c) CGA based x-vector system achieves the state-of-the-art performance on Voxceleb1 dataset.

The rest of the paper is organized as follows. Section II gives a brief introduction to the speaker embedding extractor and multi-head attention mechanism. Section 3 describes the proposed CGA in detail. Experimental setup including database description, training paradigm and results analysis are described in Section 4. Section 5 concludes the paper.

II. Related Works

A. X-vector Extractor

Since the architecture of x-vector system [12] has proved to be an effective speaker embedding extractor, this work is built on the same extractor as in [12]. The extractor employs five time-delay neural network (TDNN) layers to produce the frame-level features. The frame-level features are aggregated into a utterance-level representation through a statistics pooling layer. In this layer, the mean and standard deviation of these frame-level features are calculated and then concatenated. Two additional feedforward layers followed with a softmax layer are used to predict speaker identities. Once the network is trained, the output of the last hidden layer is regarded as the x-vector.

B. Multi-head Attention Mechanism

In speaker verification task, self attention mechanism has been employed to emphasize the frame-level features with strong speaker-discriminative information by calculating an attention weight for each frame in [13], [14]. The utterance-level representation extracted by vanilla self-attention mechanism focuses on a specific encoded representation subspace of the input utterance [15]. In other words, the utterance-level feature only reflects one aspect of input utterance. To address this problem, multi-head attention was proposed to split the encoded representations into multiple homogeneous sub-vectors called heads [16].

Formally, assuming $\mathbf{H} \in \mathbb{R}^{T\times D}$ is a frame-level feature sequence of an utterance, where T is the number of total frames in the utterance, and D is the feature dimension of each frame. The multi-head attention takes the frame-level features $\mathbf{H}$ as input, and computes the normalized weights matrix $\mathbf{A}$ as follows:

$$\mathbf{A} = \textit{softmax}(f(\mathbf{H}\mathbf{W_1})\mathbf{W_2}) \tag{1}$$

where the scalar weight $\mathbf{A} = \{\mathbf{a}_n\}_{n=1}^{N} \in \mathbb{R}^{T\times N}$, with N heads, and each column vector $\mathbf{a}_n \in \mathbb{R}^{T\times 1}$ corresponds to the attention weight vector obtained from head n. $\mathbf{W}_1$ is an intermediate linear projection layer with size of $D \times M$. $f(\cdot)$ is a activation function. $\mathbf{W}_2$ is a trainable matrix of size $M \times N$. It is noted that the *softmax*$(\cdot)$ is performed along the column. The weighted mean vectors $\mathbf{E} = \{\mathbf{e}_n\}_{n=1}^{N} \in \mathbb{R}^{N\times D}$ is obtained by:

$$\mathbf{E} = \mathbf{A}^\top\mathbf{H} \tag{2}$$

III. Context-adaptive Gaussian attention

In this section, we present the proposed context-adaptive Gaussian attention in detail. Figure 1 illustrates the conception of the proposed method. It consists of two main components: Gaussianization of attention weights and a Gaussian clustering (GC) algorithm.

A. Gaussianization of Attention Weights

Given a sequence of frame-level features $\mathbf{H}$, to obtain the utterance-level representation, a natural aggregation way is to employ attention mechanism (e.g. self-attention, multi-head attention) to assign a weight for each frame-level feature $\mathbf{h}_t$. Then, the utterance-level representation can be produced by a weighted statistic pooling. In this kind of pooling strategy, the weight of each frame-level feature is computed individually. However, it has been noted that, the context information cannot be ignored in audio processing. To capture the context information, we introduce Gaussian distributions to model the temporal correlation between adjacent frame-level features. Specifically, we compute the multi-head continuous relevance score matrix $\hat{\mathbf{A}} = \{\hat{\mathbf{a}}_n\}_{n=1}^{N} \in \mathbb{R}^{T\times N}$ across the frames as:

$$\begin{aligned} \hat{\mathbf{a}}_n[i] &= \frac{1}{\sqrt{2\pi}\sigma}\exp(-\frac{(i-c_n)^2}{2\sigma^2}) \\ c_n &= \arg\max_i(\mathbf{a}_n[i]) \quad i \in \{0,1,...,T\} \end{aligned} \tag{3}$$

where the elements of $\hat{\mathbf{a}}_n$ follow a Gaussian distribution with mean value (index of the maximum value in $\hat{\mathbf{a}}_n$) c_n, the standard deviation σ which controls the context length is fixed to a constant in consideration of the duration of the training segments. According to Eq. 3, the attention weights produced by the multi-head attention are calibrated using N Gaussian distributions respectively.

During the training stage, each of the Gaussian distribution tends to focus on a local region around the most relevant frame. So that each head models the context information of a specific sub-segment in an input utterance. Gaussianization of the attention weights derived from all the heads facilitates the SV model to capture the context information of different sub-segments in an input utterance. Unlike the traditional MHA which weights each frame-level feature individually, the proposed method takes the context information into consideration when assigning weights to different frames.

B. Gaussian Clustering

With a standard deviation constant in Gaussian distribution, the context is fixed to a certain duration. However, if Gaussian distributions between different heads overlap with each other, it suggests that each of them alone may not have the capability to model information spanning a long context. To address this issue, we propose a novel Gaussian Clustering algorithm to merge those overlapped distributions and enable the capture of the multi-span context, as shown in Figure 1.

Given two Gaussian-revised attention weights a_p and a_q with center location of c_p and c_q, respectively, if the distance

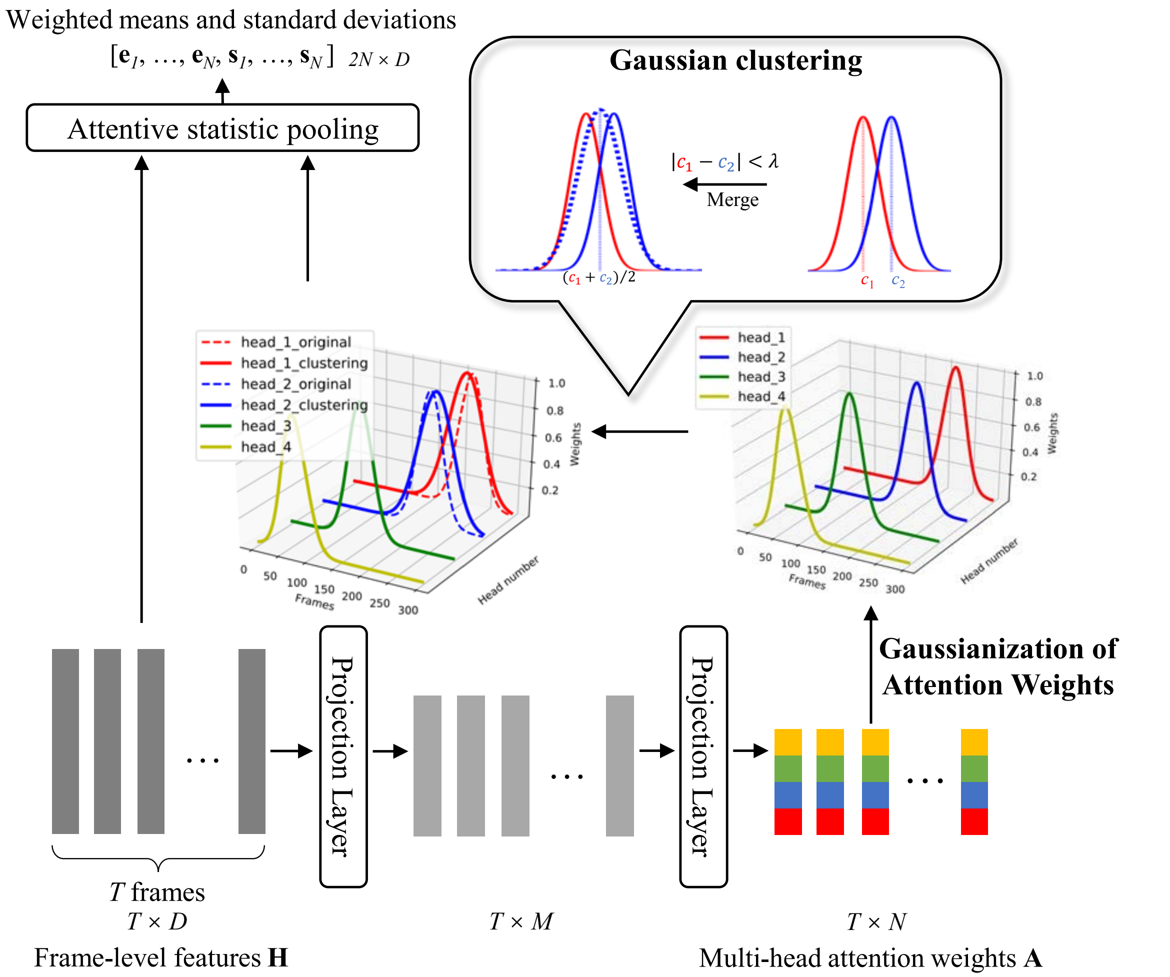

Fig. 1. Conception of our proposed context-adaptive Gaussian attention (CGA). It consists of two components: (1) Gaussian attention weight, each head's attention computed by multi-head attention is reformulated by the corresponding Gaussian distribution. (2) Gaussian Clustering: the overlapped Gaussian distributions (blue and red) are merged to a Gaussian distribution with longer context.

between two center locations is smaller than a pre-defined threshold λ, we merge these two distributions into one distribution. The new distribution is shown as follows:

$$\begin{aligned} \hat{\mathbf{a}}_n[i] &= \frac{1}{\sqrt{2\pi}\grave{\sigma}} \exp(-\frac{(i-\grave{c}_n)^2}{2\grave{\sigma}^2}) \\ \grave{c}_n &= \frac{c_p + c_q}{2}, \quad \grave{\sigma} = 2\sigma \\ i &\in \{0, 1, ..., T\} \end{aligned} \tag{4}$$

Through the attentive statistic pooling layer [13], the weighted mean vectors $\mathbf{e}_n \in \mathbb{R}^{1\times D}$ and the weighted standard deviation vectors $\mathbf{s}_n \in \mathbb{R}^{1\times D}$ of head n are computed as follows:

$$\begin{aligned} \mathbf{e_n} &= \hat{\mathbf{a}}_\mathbf{n}^\top \mathbf{H} \\ \mathbf{s_n} &= \sqrt{\hat{\mathbf{a}}_\mathbf{n}^\top \mathbf{H} \odot \mathbf{H} - \mathbf{e}_n \odot \mathbf{e}_n} \end{aligned} \tag{5}$$

The final utterance-level representation is then produced by the concatenation of the weighted mean vectors and standard deviation vectors from all the heads.

IV. EXPERIMENT AND ANALYSIS

A. Dataset

We evaluate the performance of our SV system on the VoxCeleb1 dataset [17] since it is a widely used challenging public speaker verification dataset. Specifically, this dataset consists of about 150,000 utterances from 1251 different speakers. The utterances are collected from YouTube videos. The speakers belong to different races and have a wide range of accents.

B. Implementation details

In order to compare experimental results and evaluate the performance of our proposed CGA equitably, our experimental settings are kept consistent with those of baselines [19]. Except for the pooling strategy., we utilize the same network structure, data processing, loss function, training and testing strategies in our experiments as those used in [12], [19].

Network structure: The network is constructed based on the x-vector system [12]. To be specific, a 5-layer TDNN is

TABLE I
COMPARSION OF THE PROPOSED AND STATE-OF-THE-ART APPROACHES ON VOXCELEB1 DATASET. LDE DENOTES THE LEARNABLE DICTIONARY ENCODING LAYER. SPE DENOTES THE SPATIAL PYRAMID ENCODING. MHA DENOTES MULTI-HEAD ATTENTION.(LOWER IS BETTER)

Front-end model	**Pooling Strategy**	**Loss Function**	**EER(%)**
ivector+PLDA [17]	-	-	8.8
VGG-M [17]	Temporal Average	Contrastive loss	7.8
x-vector [2]	Statistics Pooling	Softmax loss	6.0
x-vector [13]	Self-Attention	Softmax loss	4.52
VGG(1d) [2]	Statistics Pooling	Softmax loss	5.3
LSTM [3]	Self-Attention	GE2E	6.2
LSTM(head:5) [3]	MHA	GE2E	5.2
ResNet-34 [5]	Temporal Average	A-softmax	4.46
ResNet-34 [5]	LDE	A-softmax	4.56
ResNet-34 [5]	Self-Attention	A-softmax + GNLL	4.40
ResNet-34 [18]	SPE	Softmax loss	4.20
VGG-ASR [15]	Statistics Pooling	Softmax loss	4.9
VGG-ASR [15]	Self-Attention	Softmax loss	4.71
x-vector [our implementation]	Statistics Pooling	Softmax loss	5.69
x-vector [our implementation]	Self-Attention	Softmax loss	4.99
x-vector [our implementation]	MHA	Softmax loss	4.24
x-vector [proposed]	CGA	Softmax loss	4.06

TABLE II
PERFORMANCE COMPARISON OF THE CGA WITH TRADITIONAL MULTI-HEAD ATTENTION ON VOXCELEB1 FOR DIFFERENT HEAD NUMBERS (LOWER IS BETTER).

Pooling strategy	**n_head**	**EER(%)**
Multi-head Attention	4	4.43
	6	4.44
	8	4.41
	16	4.24
CGA	4	4.13
	6	4.09
	8	4.08
	16	**4.06**

TABLE III
PERFORMANCE COMPARISON OF DIFFERENT HYPERPARAMETERS IN X-VECTOR+CGA (THE HEAD NUMBER IS SET TO 4).

hyperparameters	**EER(%)**
$\sigma = 5$	4.30
$\sigma = 8$	4.39
$\sigma = 10$	4.13
$\sigma = 12$	4.30
$\sigma = 15$	4.37

used to produce frame-level features. Followed [19], [20], the kernel size for each layer is [5,5,7,1,1] without dilation.

Features: The acoustic features are 30-dimensional MFCCs with a frame length of 25ms. Mean-normalization was performed on each feature dimension of the MFCCs. Also, an energy-based voice active detection (VAD) was used to detect speech frames. To increase the diversity of the training data, we augmented the training data using reverberation and additive noises from MUSAN [21] and RIR [22], respectively.

Training: We randomly chose 64 speakers in every training step. The number of the features extracted from the truncated training speech segments ranges from 200 to 400. The network was optimized by stochastic gradient descent (SGD) with an initial learning rate 0.01. L2-regularization was employed to prevent overfitting during the training. According to the duration of the training segments, the parameters σ and λ were set to 10 based on cross validation. This means that each Gaussian distribution can model a specific sub-segment with 225ms (2* 10 * 10ms (frame shift) + 25ms (frame length)).

C. Comparison with Recent Relevant Methods

System performance comparison between the proposed CGA and the state-of-the-art relevant methods is shown in the Table I. Our x-vector+CGA system achieves the comparable lower EER among all the competitive systems. With the same x-vector feature extractor, the x-vector+MHA outperforms the original x-vector+Statistics Pooling by a relative 22.49% EER reduction. This confirms the effectiveness of the attention mechanism. By replacing the MHA with CGA, a further improvement is achieved (EER of 4.06% vs 4.24%).

D. Comparison with Traditional Multi-head Attention

In this section, we evaluate the performance of the proposed CGA and the traditional multi-head attention under different conditions of head number, i.e., 4, 6, 8, 16. The results are shown in Table II. Compared with x-vector+MHA, the x-vector+CGA achieves 6.77%, 7.88%, 7.25% and 4.24% relative improvements with 4 ,6 ,8 and 16 head. This indicates that it is reasonable to consider the context information to weight frames rather than calculate the weight of each frame separately. With the increment of head number, we notice that the performance improvement of x-vector+CGA is smoother than that of MHA. The reason is that the Gaussian clustering algorithm merges the overlapped Gaussian distribution with the increasing number of head, leading to a more stable performance.

E. Effects of hyper-parameter

The main hyper-parameter in CGA is the variance σ, which controls the context length in attention weights Gaussianization algorithm. In Table III, we evaluate the performance of x-vector+CGA(head:4) with σ varies from 5 to 15. From the Table, with the increase of σ, the performance of the system shows an unstable change. When σ is set to 4, the system has the best performance (EER: 4.13%).

V. Conclusion

This paper proposes a context-adaptive Gaussian attention (CGA) mechanism to model context information in speech utterances. CGA utilizes a set of learnable temporal Gaussian distributions to dynamically capture the highly relevant regions. To model information spanning a long context, a Gaussian Clustering algorithm is proposed to merge those overlapped distributions. In this way, the CGA is able to capture multi-span context information. Experimental results on Voxceleb1 dataset demonstrate the effectiveness of our proposed CGA mechanism.

Acknowledgment

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

References

[1] Najim Dehak, Patrick J Kenny, Réda Dehak, Pierre Dumouchel, and Pierre Ouellet. Front-end factor analysis for speaker verification. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(4):788–798, 2011.

[2] Suwon Shon, Hao Tang, and James Glass. Frame-level speaker embeddings for text-independent speaker recognition and analysis of end-to-end model. In *2018 IEEE Spoken Language Technology Workshop (SLT)*, pages 1007–1013. IEEE, 2018.

[3] Bin Liu, Shuai Nie, Yaping Zhang, Shan Liang, and Wenju Liu. Deep segment attentive embedding for duration robust speaker verification. *arXiv preprint arXiv:1811.00883*, 2018.

[4] Chunlei Zhang and Kazuhito Koishida. End-to-end text-independent speaker verification with triplet loss on short utterances. In *Interspeech*, pages 1487–1491, 2017.

[5] Weicheng Cai, Jinkun Chen, and Ming Li. Exploring the encoding layer and loss function in end-to-end speaker and language recognition system. In *Proc. Odyssey 2018 The Speaker and Language Recognition Workshop*, pages 74–81, 2018.

[6] Rongjin Li, Na Li, Deyi Tuo, Meng Yu, Dan Su, and Dong Yu. Boundary discriminative large margin cosine loss for text-independent speaker verification. In *2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6321–6325. IEEE, 2019.

[7] Junyi Peng, Rongzhi Gu, Yuexian Zou, and Wenwu Wang. Speaker-discriminative embedding learning via affinity matrix for short utterance speaker verification. In *2019 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*. IEEE, 2019.

[8] Ehsan Variani, Xin Lei, Erik McDermott, Ignacio Lopez Moreno, and Javier Gonzalez-Dominguez. Deep neural networks for small footprint text-dependent speaker verification. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4052–4056. IEEE, 2014.

[9] Georg Heigold, Ignacio Moreno, Samy Bengio, and Noam Shazeer. End-to-end text-dependent speaker verification. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5115–5119. IEEE, 2016.

[10] Li Wan, Quan Wang, Alan Papir, and Ignacio Lopez Moreno. Generalized end-to-end loss for speaker verification. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4879–4883. IEEE, 2018.

[11] David Snyder, Daniel Garcia-Romero, Daniel Povey, and Sanjeev Khudanpur. Deep neural network embeddings for text-independent speaker verification. In *Interspeech*, pages 999–1003, 2017.

[12] David Snyder, Daniel Garcia-Romero, Gregory Sell, Daniel Povey, and Sanjeev Khudanpur. X-vectors: Robust dnn embeddings for speaker recognition. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5329–5333. IEEE, 2018.

[13] Koji Okabe, Takafumi Koshinaka, and Koichi Shinoda. Attentive statistics pooling for deep speaker embedding. *Proc. Interspeech 2018*, pages 2252–2256, 2018.

[14] Yingke Zhu, Tom Ko, David Snyder, Brian Mak, and Daniel Povey. Self-attentive speaker embeddings for text-independent speaker verification. In *Interspeech*, pages 3573–3577, 2018.

[15] Miquel India, Pooyan Safari, and Javier Hernando. Self multi-head attention for speaker recognition. *Proc. Interspeech 2019*, pages 15–19, 2019.

[16] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[17] Arsha Nagrani, Joon Son Chung, and Andrew Zisserman. Voxceleb: A large-scale speaker identification dataset. *Proc. Interspeech 2017*, pages 2616–2620, 2017.

[18] Youngmoon Jung, Younggwan Kim, Hyungjun Lim, Yeunju Choi, and Hoirin Kim. Spatial pyramid encoding with convex length normalization for text-independent speaker verification. *arXiv preprint arXiv:1906.08333*, 2019.

[19] Hossein Zeinali, Lukas Burget, Johan Rohdin, Themos Stafylakis, and Jan Honza 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)*, pages 6141–6145. IEEE, 2019.

[20] Yi Liu, Liang He, and Jia Liu. Large margin softmax loss for speaker verification. *arXiv preprint arXiv:1904.03479*, 2019.

[21] David Snyder, Guoguo Chen, and Daniel Povey. Musan: A music, speech, and noise corpus. *arXiv preprint arXiv:1510.08484*, 2015.

[22] Tom Ko, Vijayaditya Peddinti, Daniel Povey, Michael L Seltzer, and Sanjeev 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)*, pages 5220–5224. IEEE, 2017.