# MULTI-VIEW SELF-ATTENTION BASED TRANSFORMER FOR SPEAKER RECOGNITION

*Rui Wang*[1,†], *Junyi Ao*[2,3,†], *Long Zhou*[4], *Shujie Liu*[4], *Zhihua Wei*[1], *Tom Ko*[2], *Qing Li*[3], *Yu Zhang*[2]

[1]Department of Computer Science and Technology, Tongji University
[2]Department of Computer Science and Engineering, Southern University of Science and Technology
[3]Department of Computing, The Hong Kong Polytechnic University
[4]Microsoft Research Asia

## ABSTRACT

Initially developed for natural language processing (NLP), Transformer model is now widely used for speech processing tasks such as speaker recognition, due to its powerful sequence modeling capabilities. However, conventional self-attention mechanisms are originally designed for modeling textual sequence without considering the characteristics of speech and speaker modeling. Besides, different Transformer variants for speaker recognition have not been well studied. In this work, we propose a novel multi-view self-attention mechanism and present an empirical study of different Transformer variants with or without the proposed attention mechanism for speaker recognition. Specifically, to balance the capabilities of capturing global dependencies and modeling the locality, we propose a multi-view self-attention mechanism for speaker Transformer, in which different attention heads can attend to different ranges of the receptive field. Furthermore, we introduce and compare five Transformer variants with different network architectures, embedding locations, and pooling methods to learn speaker embeddings. Experimental results on the VoxCeleb1 and VoxCeleb2 datasets show that the proposed multi-view self-attention mechanism achieves improvement in the performance of speaker recognition, and the proposed speaker Transformer network attains excellent results compared with state-of-the-art models.

***Index Terms—*** speaker recognition, Transformer, speaker identification, speaker verification.

## 1. INTRODUCTION

Transformer models [1] have recently demonstrated exemplary performance on a broad range of natural language processing (NLP) tasks, such as machine translation and question answering. Compared with recurrent neural networks (RNNs) and convolutional neural networks (CNNs), the advantage of self-attention in Transformer lies in its high parallelization capabilities and global modeling capabilities. Recently, there have been increasing interests in exploring Transformers for spoken language processing, e.g., speech recognition [2, 3, 4], speech synthesis [5, 6], and speaker recognition [7, 8].

In speaker recognition, however, convolutional architectures remain dominant, such as residual network (ResNet) [9, 10, 11] and time delay neural network (TDNN) [12, 13]. Inspired by the successes of self-attention in NLP, several works have tried to combine CNN-like architectures with self-attention by either replacing utterance-level pooling layers or frame-level convolutions blocks [14, 15, 16]. Nevertheless, the overall structure of the previous work remains unchanged, and the application of Transformer to speaker recognition is limited. It is an interesting topic to explore effective ways of modeling speaker embedding with Transformer.

Applying Transformer to speaker tasks has two challenges: 1) Transformer is hard to be scaled efficiently since acoustic features sequences are much longer than text sentences. 2) Transformer is deficient in some of the inductive biases inherent to CNNs, such as translation equivalence and locality [17]. To enable Transformers to model the long-duration speech and locality, we propose a multi-view self-attention mechanism, where different attention heads can attend to different ranges of the receptive field to boost the capabilities of capturing global dependencies and modeling the locality. Furthermore, we present a thorough empirical exploration of different Transformer models with different network architectures, embedding locations, and pooling methods for speaker recognition, equipped with our proposed multi-view self-attention mechanism. We train the Transformer to represent speakers in a supervised speaker classification fashion, which encourages the encoder to capture different speaker properties by short-term or long-term dependencies. Experiments on the VoxCeleb1 and VoxCeleb2 datasets show that the proposed speaker Transformer network outperforms other CNNs and Transformer-based networks in that it achieves 96.38% top-1 accuracy on the identification task and 2.56% equal error rate on the verification task.

Our contributions can be summarized as follows. (1) We propose a multi-view self-attention mechanism for Transformer-based speaker networks, which enable to capture global dependencies and model the locality. (2) We study the proposed multi-view self-attention mechanism in different Transformer variants with different network structures, embedding locations, and pooling methods.

## 2. RELATED WORK

Transformers were proposed by Vaswani et al. [1] for machine translation, and have become the state-of-the-art method in many NLP tasks. To apply Transformers in the context of speaker recognition, several works study this issue.

For speaker recognition, the attention mechanism has been studied with the pooling mechanism as an alternative to aggregate temporal information. Cai et al. [9] introduce a self-attentive pooling layer to obtain the utterance-level representation. Okabe et al. [13] propose attentive statistics pooling, which gives different weights to different frames and generates weighted means and standard deviations. Wu et al. [18] improve it by adopting a vectorial attention mechanism. India et al. [14] present double multi-head attention pooling, where an additional self-attention layer is added to the pooling layer to enhance the attentive pooling mechanism. To improve

[†]Equal contribution. Work done during internship at Microsoft Research Asia.

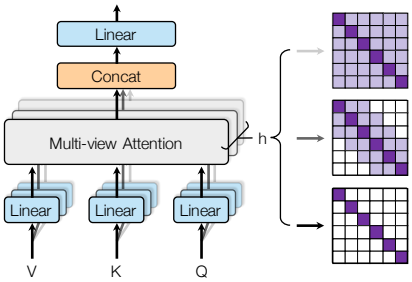



**Fig. 1**: The proposed multi-view self-attention mechanism.

the diversity of attention heads, Wang et al. [19] propose multi-resolution multi-head attention pooling, which incorporates different resolutions of attentive weights. Instead of using a fixed query for all utterances, Zhu et al. [15] introduce a self-attention mechanism with an input-aware query to consider overall information and speech dynamics over each utterance.

On the other hand, Jiang et al. [20] introduce a channel-wise attention mechanism as a gate, which can exploit global time-frequency information to improve the sensitivity of informative features while suppressing less useful ones. Yu et al. [21] propose a dynamic channel-wise selection mechanism based on the softmax attention to gather effective information and estimate the importance of network branches. These works utilize the attention mechanism as a selection of channel-wise information in a block of the feature extractor. It is of limited worth for extracting speaker embedding.

Most recently, the attention layer has been directly stacked as a part of layers or the whole feature extractor. On the top of the x-vectors framework [12], Shi et al. [16] apply Transformer encoders to both frame-level and segment-level to capture features at different scales. Zhu et al. [15] propose a serialized multi-layer multi-head attention to obtain the final utterance-level embedding by aggregating the utterance-level vectors from all heads. These works depend on the frame-level sophisticated convolutional networks such as TDNNBlocks [12] and SE-Res2Blocks [22]. In contrast, Safari et al. [8] propose a tandem self-attention encoding and pooling (SAEP) mechanism, which stacks two Transformer encoder layers followed by an additive attention pooling. Metilda et al. [7] propose s-vectors which stacks Transformer encoder layers followed by a statistics pooling layer and two linear layers. However, Transformer-based feature extractors lack the capacity to model the locality and possess inferior performance in speaker recognition.

## 3. SPEAKER TRANSFORMER

### 3.1. Multi-View Self-Attention

We propose a multi-view self-attention mechanism for Transformer to enhance the capabilities of capturing global dependencies while modeling the locality. As shown in Fig 1, the multi-view self-attention mechanism is implemented as self-attention with sliding windows of different sizes, in which each attention head has a different range of the receptive field.

Given the importance of local context, the proposed multi-view self-attention mechanism employs windows with different sizes surrounding each token. Using multiple stacked layers of such windowed attention creates various receptive fields, where top layers have access to long-range input locations. Therefore, similar to CNNs, it can build representations that incorporate information across the input. Specifically, given a fixed window size $w$, each token attends to $\frac{1}{2}w$ tokens on both sides. At the $l$-th layer of a Transformer encoder, the receptive field size ranges from $l \times w_{\text{min}}$ to $l \times w_{\text{max}}$, where $w_{\text{min}}$ and $w_{\text{max}}$ are the minimum and maximum window sizes for all layers, respectively.

For different Transformer variants, it might be helpful to use different values of $w_{\text{min}}$ and $w_{\text{max}}$ for each layer to model long-term or short-term dependencies. However, it is computationally prohibitive to fine-tune the size of windows at each layer, because there is a vast search space of window size as the temporal length and layer number increase. Intuitively, we simplify the selection of sliding window for the $i$-th head at the $l$-th layer to explicitly model different ranges of receptive fields by setting them as

$$w_i^l = \begin{cases} 2^i + 1, & \text{if } i \geq 1 \\ 1, & i = 0 \end{cases}.$$

Given the matrices $Q$, $K$, and $V$ in the Transformer model [1], the proposed multi-view attention mechanism is formulated as

$$\text{Attention}(Q, K, V) = M \odot \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V, \qquad (1)$$

where $M \in \mathbb{R}^{B \times H \times N \times N}$ is a head-wise masking matrix, $B$ is batch size, $H$ is the number of heads, $N$ is the number of steps, and $d_k$ is the dimension of queries and keys as mentioned in [1].

### 3.2. Transformer Variants

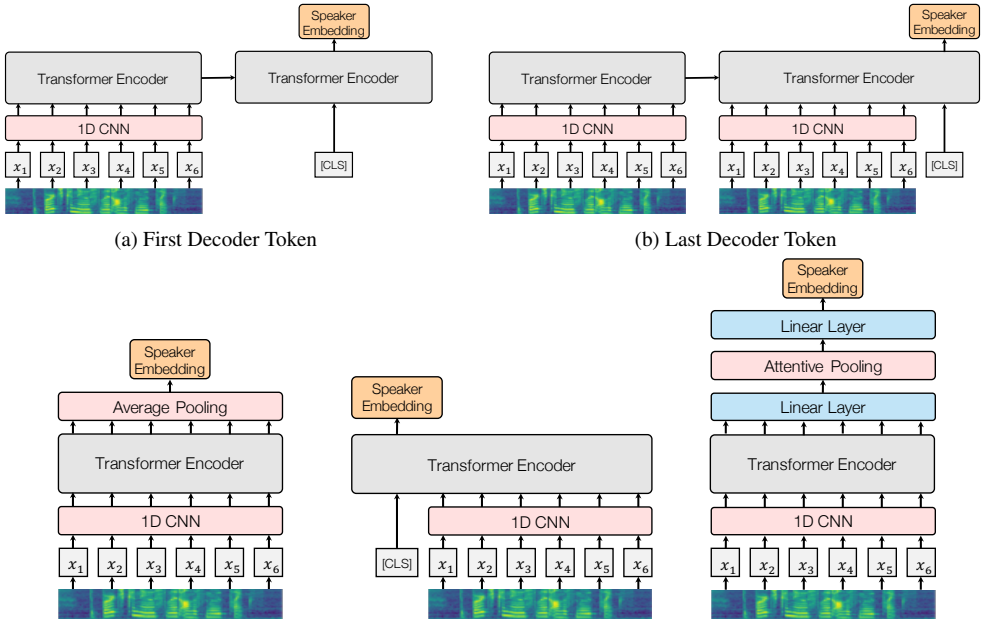
The general Transformer architecture used in machine translation [1] consists of encoder blocks and decoder blocks. Each encoder block contains a multi-head attention and a feedforward network, while each decoder block has an additionally masked multi-head attention. All of the attention modules and feedforward networks are in conjunction with the residual connection and layer normalization.

We study five variants of the Transformer architecture for identifying speakers and extracting speaker embedding. We use an architecture with a 6-layer encoder, a 3-layer decoder, 512 attention size, 2048 hidden size, and 8 attention heads, which contain parameters up to 34.6 million. For all variants, the input $\mathbf{X}$ is firstly processed by two one-dimensional convolutional layers (called subsample encoder prenet) for downsampling to a quarter of the input length, followed by the Transformer encoder. Downsampling acoustic features $\mathbf{H}$ accelerates the processing efficiency of a speech utterance while forming coarse features to lay the base of extracting speaker-discriminative characteristics. In the following, we introduce the five variants.

(a) **First Decoder Token**. As shown in Fig 2a, the Transformer architecture with an encoder and a decoder is considered as the speaker network. Specifically, the Transformer decoder and encoder take the `[CLS]` token and $\mathbf{H}$ as the input, respectively. In this variant, the Transformer decoder acts as a multi-layer multi-head attentive pooling and takes advantage of stacked pooling, which is helpful to generate speaker-discriminative vectors.

(b) **Last Decoder Token**. Similar to [23], we formulate the problem of speaker classification as a sequence classification task. Different from the first variant, $\mathbf{H}$ is both fed into the encoder and decoder. The decoder can be considered as input-wise pooling. Then the final step of the decoder takes `[CLS]` token as input and generates speaker embedding, as shown in Fig 2b.

(c) **Average Encoder Token**. The naive temporal average pooling is directly applied without frame-level and utterance-level transformations to represent speakers. The output of the Transformer encoder is averaged to obtain speaker embedding as shown in Fig 2c.




(a) First Decoder Token (b) Last Decoder Token

(c) Average Encoder Tokens (d) First Encoder Token (e) Pooling Encoder Tokens

**Fig. 2**: Five Transformer variants for extracting speaker embedding.

(d) **First Encoder Token**. As shown in Fig 2d, `[CLS]` token is concatenated with $\mathbf{H}$ as the input of the encoder, and the first output of the Transformer encoder is regarded as speaker embedding. Compared with other variants that utilize all tokens of the encoder, this variant utilizes a single token at the top layer of the encoder, which might cause the reducing diversity of temporal information.

(e) **Pooling Encoder Tokens**. Following the architectural setting in x-vector [12], a linear layer, attentive pooling, and multiple hidden layers are sequentially stacked on the Transformer encoder, as shown in Fig 2e. The difference between the x-vector and this Transformer variant is that the former uses a TDNN network as the feature extractor. Compared with [7, 8], the linear transformation of inputs and temporal pooling are replaced by the sub-sample encoder prenet and attentive statistics pooling, respectively.

## 4. EXPERIMENTS

### 4.1. Setup

**Dataset.** We focus on text-independent speaker recognition and use the VoxCeleb dataset to evaluate the performance on both speaker identification and verification tasks. The VoxCeleb dataset, containing VoxCeleb1 [24] and VoxCeleb2 [25], is a large-scale text-independent speaker recognition dataset collected "in the wild". The VoxCeleb1 has over 100,000 utterances from 1,251 celebrities, while the VoxCeleb2 has over 1,000,000 utterances from 6,112 celebrities. We use the official split of VoxCeleb1 for the speaker identification task, where the test set contains 8,251 utterances from these 1,251 celebrities. For the speaker verification task, we consider two settings. The VoxCeleb1 with 1,211 speakers and VoxCeleb2 with 5,994 speakers are used for training, respectively. The test set contains 4,715 utterances from 40 speakers in VoxCeleb1. There are 37,720 pairs of trials, including 18,860 target pairs.

**Acoustic Features.** In our experiments, the *librosa* toolkit is used to extract 80-dimensional mel-filter banks with the 64ms window and 16ms shift to represent the speech signal. No data augmentation is used during the training or test process. All features are subject to 200-frame utterance-level cepstral mean variance normalization. We apply SpecAugment [26] which randomly masks 0 to 20 frames up to twice in the time domain and 0 to 10 frequency banks up to twice.

**Training and Metrics.** Using *fairseq* [27], we train all models on both tasks via the Adam optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.999$, a weight decay of 0.1, a batch of 2048, the inputs of 200 frames, and the dropout of 0.1. We adjust the learning rate based on a 60k-step cycle of a triangular cyclical schedule between $10^{-8}$ and $5 \times 10^{-4}$. Four cycles are applied to VoxCeleb2 while 2 cycles is to VoxCeleb1 due to a smaller data size. During the test stage, speaker embeddings are extracted from the whole speech signal. We report the top-1 accuracy (ACC) as the identification metric and the equal error rate (EER) for verification.

### 4.2. Different Transformer Variants

The results of five variants with or without multi-view self-attention (MV) are shown in Table 1, where ACC on VoxCeleb1 and EER on VoxCeleb1 and VoxCeleb2 are reported. Except variant (d), most Transformer variants with MV outperform those without MV, which indicates the proposed MV is helpful to improve the performance. Specifically, training on middle-scale corpus VoxCeleb1, MV achieves improvements in most tasks for variants (a), (b), (c), and (e) but slightly degrades in the verification task for variant (a) and identification task for variant (c). As the size of the dataset scales, the performance of speaker embedding boosts consistently and attains 5.9%-10.3% improvement for variants (a), (b), (c), and (e). It suggests that MV is an effective self-attention enhancement technique that can be jointly used with other techniques such as attentive statistics pooling in variant (e).

On the other hand, the performance of variant (d) with MV significantly degrades regardless of the scale of datasets varies. This variant uses the first token at the top layer of the encoder as speaker embedding. The self-attention with different sliding window masks forces heads on the importance of various local context, which

**Table 1**: Performance of MV-based Transformer on VoxCeleb1.

| Variants | ACC (%) ↑ | | EER (%) ↓ | | EER (%) ↓ | |
|---|---|---|---|---|---|---|
| | Vox1 | + MV | Vox1 | + MV | Vox2 | + MV |
| (a) | 94.33 | **94.36** | **5.33** | 5.45 | 2.72 | **2.56** |
| (b) | 93.61 | **94.09** | 5.89 | **5.40** | 2.92 | **2.68** |
| (c) | **92.96** | 91.81 | 6.33 | **6.13** | 3.60 | **3.23** |
| (d) | **92.29** | 88.16 | **5.96** | 7.37 | **3.32** | 3.96 |
| (e) | 95.04 | **96.38** | 4.77 | **4.35** | 2.89 | **2.68** |

makes heads learn different distributions of temporal information and contribute unequally to extract speaker embedding. It causes that variant (d) suffers the loss of part of the temporal information.

Except variant (d), the Transformer variants with or without MV give the same ranking of the verification performance on the VoxCeleb2, i.e., (a), (e), (b), (c). Variant (c) introduces a naive temporal average pooling among features derived from the Transformer encoder, which represents the fundamental capability of identifying the speaker. On the top of the transformer encoder, variant (a) applies multi-head attentive pooling, variant (e) applies attentive statistics pooling, and variant (c) applies input-wise pooling to achieve superior performance. Although variant (b) provides an additional input, it is inferior to variant (a). It is probably that the additional input causes over-regularization to the decoder, whose weights require learning the mapping function from both the input and `[CLS]` token to the speaker identity. By considering that the temporal standard in the utterance-level features is complementary to the temporal average pooling, it implies that combining multi-head attentive pooling and statistics pooling can further improve the performance.

### 4.3. Comparison on VoxCeleb

We compare the proposed method with several models, including VGG [24], TDNN [12, 13], ResNet [9, 10, 11], and Transformer [7, 8]. According to results on the VoxCeleb speaker recognition tasks shown in Table 2, we can see that the proposed method outperforms most works using convolutional network or attention mechanism on three speaker tasks. Compared with those methods based on VGG, TDNN, and ResNet as the feature extractor, the proposed Transformer encoder stacked on a sub-sample prenet attains excellent performance in both ACC and EER. To the best of our knowledge, the obtained ACC is the state-of-the-art performance, which indicates that the Transformer-based speaker network possesses superior capability for classification. On the other hand, our work is inferior to the TDNN with attentive statistics pooling [13]. It is probably that the data augmentation technique increases the diversity of training dataset, which is helpful to generalize to unseen speakers and unseen acoustic scenes.

Regarding the feature extractor, variant (a) outperforms the VGG with multi-head attention [14], and variant (e) achieves comparable performance with the TDNN equipped with attentive statistics pooling [13]. It suggests that compared with several popular CNNs, the Transformer encoder has a comparative capability to extract frame-level features for generating speaker-discriminative embeddings.

For the Transformer-based speaker verification, the proposed method achieves EER of 4.35% and 2.56% when training on the VoxCeleb1 and VoxCeleb2, respectively. We further boost the performance based on [7, 8] where the Transformer encoder is applied to extract features. For example, considering that SEAP [7] designs a lightweight network with 1.60 million parameters, one reason for the improvement of the proposed method is the model scaling. Regardless of the small size, it often does not scale effectively as the length of inputs increases. Therefore, the sub-sample prenet is employed in our work, which leads to a significant reduction in terms of storage size, processing, and memory. S-vectors [8] trained on the VoxCeleb1 and VoxCeleb2 development sets and data augmentation is inferior to the proposed method. It suggests several architectural enhancements to the Transformer-based speaker network such as attentive pooling, multi-head pooling, and multi-layer pooling.

**Table 2**: Performance comparison on the VoxCeleb1 test Set.

| Implementaion | Extractor | ACC (%) | EER (%) |
|---|---|---|---|
| **Training on VoxCeleb1 development** | | | |
| VGG-M [24] | VGG | 80.5 | 7.8 |
| X-vector [12] | TDNN | - | 7.83 |
| Atten. Stats.*[13] | TDNN | - | **3.85** |
| Cai et al. [9] | ResNet | 89.9 | 4.46 |
| Chung et al. [11] | ResNet | 89.0 | 5.26 |
| SAEP [8] | Transformer | - | 7.13 |
| S-vectors [7] | Transformer | - | 5.50 |
| Our work (e) | CNN+Transformer | **96.38** | 4.35 |
| **Training on VoxCeleb2 development** | | | |
| MHA [14] | VGG | | 3.19 |
| Atten. Stats. [13] | TDNN | | 2.59 [18] |
| Xie et al. [10] | ResNet | | 3.22 |
| SAEP [8] | Transformer | | 5.44 |
| S-vectors*$^{+}$[7] | Transformer | | 2.67 |
| Our work (a) | CNN+Transformer | | **2.56** |
| Our work (e) | CNN+Transformer | | 2.68 |

* Training using data augmentation.
$^{+}$ Training dataset includes VoxCeleb2 and VoxCeleb1 dev set.

## 5. CONCLUSION

In this work, we explore five Transformer variants for speaker recognition. A multi-view self-attention mechanism is proposed to balance the capabilities of capturing global dependencies and modeling the locality by using sliding windows with different sizes for each attention head. The proposed attention mechanism achieves improvements on most variants for both speaker identification and speaker verification tasks. Moreover, the proposed model attains excellent results compared to several previous CNN-based and Transformer-based models. Our method achieves 96.38% top-1 accuracy for the speaker identification task on Voxceleb1, which is state-of-the-art to the best of our knowledge, and 4.35% and 2.56% EER on VoxCeleb1 and VoxCeleb2, respectively, for the speaker verification task. In the future work, we will utilize larger datasets by pretraining techniques [28, 29] and employ data augmentation techniques [30, 31] to further boost the performance.

## 6. ACKNOWLEDGEMENTS

This work is partially supported by the National Nature Science Foundation of China (No. 61976160, 62076182, 61906137) and Technology research plan project of Ministry of Public and Security (Grant No. 2020JSYJD01) and Shanghai Science and Technology Plan Project (No. 21DZ1204800).

## 7. REFERENCES


[1] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, L ukasz Kaiser, and Illia Polosukhin, "Attention is all you need," in *Proc. NeurIPS*, 2017, pp. 6000–6010.

[2] Xie Chen, Yu Wu, Zhenghao Wang, Shujie Liu, and Jinyu Li, "Developing real-time streaming transformer transducer for speech recognition on large-scale dataset," in *Proc. ICASSP*, 2021, pp. 5904–5908.

[3] Long Zhou, Jinyu Li, Eric Sun, and Shujie Liu, "A configurable multilingual model is all you need to recognize all languages," *arXiv preprint arXiv:2107.05876*, 2021.

[4] Jinyu Li, "Recent advances in end-to-end automatic speech recognition," *arXiv preprint arXiv:2111.01690*, 2021.

[5] Naihan Li, Shujie Liu, Yanqing Liu, Sheng Zhao, and Ming Liu, "Neural speech synthesis with transformer network," in *Proc. AAAI*, 2019, pp. 6706–6713.

[6] Wen-Chin Huang, Tomoki Hayashi, Yi-Chiao Wu, Hirokazu Kameoka, and Tomoki Toda, "Voice Transformer Network: Sequence-to-sequence voice conversion using transformer with text-to-speech pretraining," in *Proc. Interspeech*, 2020, pp. 4676–4680.

[7] Metilda Sagaya Mary N J, Sandesh V Katta, and S Umesh, "S-vectors: Speaker embeddings based on transformer's encoder for text-independent speaker verification," *arXiv preprint arXiv:2008.04659*, 2020.

[8] Pooyan Safari, Miquel India, and Javier Hernando, "Self-attention encoding and pooling for speaker recognition," in *Proc. Interspeech*, 2020, pp. 941–945.

[9] 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, pp. 74–81.

[10] Weidi Xie, Arsha Nagrani, Joon Son Chung, and Andrew Zisserman, "Utterance-level aggregation for speaker recognition in the wild," in *Proc. ICASSP*, 2019, pp. 5791–5795.

[11] Joon Son Chung, Jaesung Huh, and Seongkyu Mun, "Delving into VoxCeleb: Environment invariant speaker recognition," in *Proc. Odyssey*, 2020, pp. 349–356.

[12] David Snyder, Daniel Garcia-Romero, Gregory Sell, Daniel Povey, and Sanjeev Khudanpur, "X-vectors: Robust DNN embeddings for speaker recognition," in *Proc. ICASSP*, 2018, pp. 5329–5333.

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

[14] Miquel India, Pooyan Safari, and Javier Hernando, "Double multi-head attention for speaker verification," in *Proc. ICASSP*, 2021, pp. 6144–6148.

[15] Hongning Zhu, Kong Aik Lee, and Haizhou Li, "Serialized multi-layer multi-head attention for neural speaker embedding," *arXiv preprint arXiv:2107.06493*, 2021.

[16] Yanpei Shi, Mingjie Chen, Qiang Huang, and Thomas Hain, "T-vectors: Weakly supervised speaker identification using hierarchical transformer model," *arXiv preprint arXiv:2010.16071*, 2020.

[17] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[18] Yanfeng Wu, Chenkai Guo, Hongcan Gao, Xiaolei Hou, and Jing Xu, "Vector-based attentive pooling for text-independent speaker verification," in *Proc. Interspeech*, 2020, pp. 936–940.

[19] Zhiming Wang, Kaisheng Yao, Xiaolong Li, and Shuo Fang, "Multi-resolution multi-head attention in deep speaker embedding," in *Proc. ICASSP*, 2020, pp. 6464–6468.

[20] Yiheng Jiang, Yan Song, Ian McLoughlin, Zhifu Gao, and Lirong Dai, "An effective deep embedding learning architecture for speaker verification," in *Proc. Interspeech*, 2019, pp. 4040–4044.

[21] Ya-Qi Yu and Wu-Jun Li, "Densely connected time delay neural network for speaker verification," in *Proc. Interspeech*, 2020, pp. 921–925.

[22] Brecht Desplanques, Jenthe Thienpondt, and Kris Demuynck, "ECAPA-TDNN: Emphasized channel attention, propagation and aggregation in TDNN based speaker verification," in *Proc. Interspeech*, 2020, pp. 3830–3834.

[23] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer, "BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," in *Proc. ACL*, 2020, pp. 7871–7880.

[24] Arsha Nagrani, Joon Son Chung, and Andrew Zisserman, "VoxCeleb: A large-scale speaker identification dataset," in *Proc. Interspeech*, 2017, pp. 2616–2620.

[25] Joon Son Chung, Arsha Nagrani, Andrew Zisserman, and Assoc Int Speech Commun, "Voxceleb2: Deep speaker recognition," in *Proc. Interspeech*, 2018, pp. 1086–1090.

[26] Daniel S. Park, William Chan, Yu Zhang, Chung Cheng Chiu, Barret Zoph, Ekin D. Cubuk, and Quoc V. Le, "Specaugment: A simple data augmentation method for automatic speech recognition," in *Proc. Interspeech*, 2019, pp. 2613–2617.

[27] Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli, "fairseq: A fast, extensible toolkit for sequence modeling," in *Proc. NAACL-HLT*, 2019.

[28] Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed, "HuBERT: Self-supervised speech representation learning by masked prediction of hidden units," *arXiv preprint arXiv:2106.07447*, 2021.

[29] Junyi Ao, Rui Wang, Long Zhou, Shujie Liu, Shuo Ren, Yu Wu, Tom Ko, Qing Li, Yu Zhang, Zhihua Wei, Yao Qiao, Jinyu Li, and Furu Wei, "Speecht5: Unified-modal encoder-decoder pre-training for spoken language processing," *arXiv preprint arXiv:2110.07205*, 2021.

[30] Tom Ko, Vijayaditya Peddinti, Daniel Povey, and Sanjeev Khudanpur, "Audio augmentation for speech recognition," in *Proc. Interspeech*, 2015, pp. 3586–3589.

[31] 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 *Proc. ICASSP*, 2017, pp. 5220–5224.