한국음향학회지 제39권 제5호 pp. 497~504 (2020)
The Journal of the Acoustical Society of Korea Vol.39, No.5 (2020)
https://doi.org/10.7776/ASK.2020.39.5.497

pISSN : 1225-4428
eISSN : 2287-3775

Masked cross self-attentive encoding based speaker embedding for speaker verification

화자 검증을 위한 마스킹된 교차 자기주의 인코딩 기반 화자 임베딩

Soonshin Seo[1] and Ji-Hwan Kim[1†]

(서순신,[1] 김지환[1†])

[1]Sogang University

(Received August 7, 2020; accepted September 9, 2020)

ABSTRACT: Constructing speaker embeddings in speaker verification is an important issue. In general, a self-attention mechanism has been applied for speaker embedding encoding. Previous studies focused on training the self-attention in a high-level layer, such as the last pooling layer. In this case, the effect of low-level layers is not well represented in the speaker embedding encoding. In this study, we propose Masked Cross Self-Attentive Encoding (MCSAE) using ResNet. It focuses on training the features of both high-level and low-level layers. Based on multi-layer aggregation, the output features of each residual layer are used for the MCSAE. In the MCSAE, the interdependence of each input features is trained by cross self-attention module. A random masking regularization module is also applied to prevent overfitting problem. The MCSAE enhances the weight of frames representing the speaker information. Then, the output features are concatenated and encoded in the speaker embedding. Therefore, a more informative speaker embedding is encoded by using the MCSAE. The experimental results showed an equal error rate of 2.63 % using the VoxCeleb1 evaluation dataset. It improved performance compared with the previous self-attentive encoding and state-of-the-art methods.
Keywords: Speaker verification, Masked cross self-attentive encoding, Speaker embedding, ResNet

PACS numbers: 43.72.Fx, 43.71.Bp

초 록: 화자 검증에서 화자 임베딩 구축은 중요한 이슈이다. 일반적으로, 화자 임베딩 인코딩을 위해 자기주의 메커니즘이 적용되어졌다. 이전의 연구는 마지막 풀링 계층과 같은 높은 수준의 계층에서 자기 주의를 학습시키는 데 중점을 두었다. 이 경우, 화자 임베딩 인코딩 시 낮은 수준의 계층의 영향이 감소한다는 단점이 있다. 본 연구에서는 잔차 네트워크를 사용하여 Masked Cross Self-Attentive Encoding(MCSAE)를 제안한다. 이는 높은 수준 및 낮은 수준 계층의 특징 학습에 중점을 둔다. 다중 계층 집합을 기반으로 각 잔차 계층의 출력 특징들이 MCSAE에 사용된다. MCSAE에서 교차 자기 주의 모듈에 의해 각 입력 특징의 상호 의존성이 학습된다. 또한 랜덤 마스킹 정규화 모듈은 오버 피팅 문제를 방지하기 위해 적용된다. MCSAE는 화자 정보를 나타내는 프레임의 가중치를 향상시킨다. 그런 다음 출력 특징들이 합쳐져 화자 임베딩으로 인코딩된다. 따라서 MCSAE를 사용하여 보다 유용한 화자 임베딩이 인코딩된다. 실험 결과, VoxCeleb1 평가 데이터 세트를 사용하여 2.63 %의 동일 오류율를 보였다. 이는 이전의 자기 주의 인코딩 및 다른 최신 방법들과 비교하여 성능이 향상되었다.
핵심용어: 화자검증, 마스킹된 교차 자기주의 인코딩, 화자 임베딩, 잔차 네트워크

†**Corresponding author:** Ji-Hwan Kim (kimjihwan@sogang.ac.kr)
Department of Computer Science and Engineering, Sogang University, 35 Baekbum-ro, Mapo-gu, Seoul 04107, Republic of Korea
(Tel: 82-2-705-8924)

I. Introduction

Speaker recognition aims to identify speaker information from input speech. A type of speaker recognition is Speaker Verification (SV). It determines whether the test speaker's speech is accept or reject compared to the enrolled speaker's speech.

Traditionally, the Gaussian mixture model with universal background model has been used to encode supervector representing speaker information.[1,2] Next, a joint factor analysis method has been proposed to separate the supervector from the channel and speaker factors.[3] However, these methods required an enormous amount of data for the enrollment. An i-vector has been proposed to solve this issue. It has been used with probabilistic linear discriminant analysis.[4-5]

Since the introduction of deep learning, d-vector have been extracted directly from Deep Neural Networks (DNN).[6] It is trained by using the DNN-based speaker classifier. Then the activations of the last hidden layer are encoded as speaker embedding. In addition, speaker embedding encodings using various DNN-based models have been proposed. In time delay neural network (TDNN)-based model, the x-vector has been proposed. It is a fixed dimensional statistics vector, encoded by using statistical pooling.[7] Among the Convolutional Neural Network (CNN)-based models, ResNet[8] has been used as a representative model for speaker embeddings.[9-14]

Attention mechanisms successfully applied to other areas, such as image and language processing.[15-19] In SV, TDNN or CNN model-based speaker embedding encodings using attention mechanism have been proposed.[9,12,20-25] Especially, the self-attention mechanism[16] has exhibited high performance in speaker embedding encoding as called Self-Attentive Pooling (SAP).[9,24,25] The SAP is used to encode frame-level features into a utterance-level feature. It focuses on the frames by training interdependence with a context vector. In addition, an SAP-derived method called Multi-Head Attentive Pooling (MHAP) has been proposed to improve performance.[25]

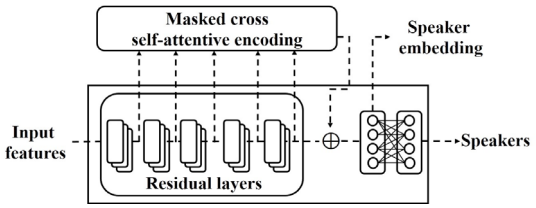

Fig. 1. Overview of the proposed network using MCSAE.

However, these previous methods are focused on training the self-attention in a high-level layer instead of the lower-level layers. In other words, speaker embedding is encoded by using only the output feature of the last pooling layer. It results in decreased low-level features effect in the encoding of a speaker embedding. Therefore, it is difficult to encode the speaker embedding with more discriminative power.

Therefore, we propose a Masked Cross Self-Attentive Encoding (MCSAE). This is a new SAP-derived speaker embedding encoding using ResNet. MCSAE focuses on the features of both the high-level and low-level layers in the self-attention training. Based on Multi-Layer Aggregation (MLA),[14] the output features of each residual layer are used as the input pair of the MCSAE, as shown in Fig. 1. In the MCSAE, the interdependence of each input features is trained by a cross self-attention module. A random masking regularization module also applied to prevent overfitting problem. The MCSAE enhances the weight of frames representing the speaker information. Then, the output features are concatenated and encoded in the speaker embedding. Therefore, a more discrimitive speaker embedding is encoded by using the MCSAE.

We introduce the concept of self-attention and its use in Section 2, describe the proposed MCSAE in Section 3, present the results in Section 4, and present our conclusions in Section 5.

II. Concept of self-attention mechanism and its use

2.1 Self-attention mechanism

The principle of the self-attention mechanism is to focus

on training the specific context information. In machine translation, self-attention using scaled dot-product attention and MHAP has been proposed.[16] The scaled dot-product attention is formulated as in Eq. (1).

$$attention\ (\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V}) = softmax\left(\frac{\boldsymbol{Q}\boldsymbol{K}^T}{\sqrt{d_{\boldsymbol{K}}}}\right)\boldsymbol{V}. \quad (1)$$

The inputs are comprised of the query ($\boldsymbol{Q}$), key ($\boldsymbol{K}$), and value ($\boldsymbol{V}$). To train the relationship between $\boldsymbol{Q}$ and $\boldsymbol{K}$, scaling is applied to compute similarity using dot product operations on all $\boldsymbol{Q}$ and $\boldsymbol{K}$ elements and each element is divided by $\sqrt{d_{\boldsymbol{K}}}$ ($d_{\boldsymbol{K}}$ is the dimension of $\boldsymbol{K}$). Next, after applying the softmax method for normalization, the weights for $\boldsymbol{V}$ are obtained. The more similar $\boldsymbol{V}$ is to $\boldsymbol{Q}$, the higher its value, more attention will be paid to $\boldsymbol{V}$.

2.2 Self-attention in speaker verification

In SV, SAP, which is applied to TDNN and ResNet-based models, outperforms both the conventional Temporal Average Pooling (TAP) and Global Average Pooling (GAP).[9,24,25]

An input feature of the hidden layer $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_l, \cdots, \boldsymbol{x}_L]$ of length L is fed into a fully-connected hidden layer to obtain $\boldsymbol{H} = [\boldsymbol{h}_1, \boldsymbol{h}_2, \cdots, \boldsymbol{h}_l, \cdots, \boldsymbol{h}_L]$. Given that $\boldsymbol{h}_l \in \mathbb{R}^d$ and a learnable context vector $\boldsymbol{u} \in \mathbb{R}^d$ the attention weight $\boldsymbol{w}_l$ is measured by training the similarity between $\boldsymbol{h}_l$ and $\boldsymbol{u}$ with softmax normalization as in Eq. (2).

$$\boldsymbol{w}_l = \frac{exp(\boldsymbol{h}_l^T \cdot \boldsymbol{u})}{\sum_{i=1}^{L} exp(\boldsymbol{h}_i^T \cdot \boldsymbol{u})}. \quad (2)$$

Then, the embedding vector $\boldsymbol{e} \in \mathbb{R}^d$ is generated by the weighted sum between the normalized attention weights $\boldsymbol{w}_l$ and $\boldsymbol{x}_l$ as in Eq. (3).

$$\boldsymbol{e} = \sum_{l=1}^{L} \boldsymbol{x}_l \boldsymbol{w}_l. \quad (3)$$

Hence, an utterance-level feature focused on each frame is encoded. Additionally, based on this process, the MHAP is conducted by performing several linear projections on each input.[25]

2.3 Previous cross attention and masking methods

Our proposed cross self-attention and masking methods are inspired by the studies conducted by the References [18], [19], respectively.

In image-text matching, cross attention has been proposed to identify the appropriate text appearing in an input image.[18] The inputs are encoded in both image-text and text-image formulations. Then, the cross attention is applied to both pairs for obtaining more accurate weights than that obtained with just one attention mechanism.

In person re-identification, masking method and attention mechanism have been applied. These are used to solve the problem of the neglected dissimilarities between the source and the target.[19] In the attention process between the source and the target, a masking matrix of integer [1 or -1], according to the label is used.

III. Masked cross self-attentive encoding based speaker embedding

3.1 Model architecture

The proposed model builds on previous research on the speaker embedding encoding based on MLA.[14] The modified ResNet model is trained for speaker classification in an end-to-end training process using several pooling layers.

The proposed model architecture is modified by using a standard ResNet-34 model.[8] It adds MCSAE after each pooling, as shown in Fig. 1 and Table 1. The proposed model has 4 residual layers, 16 residual blocks, and half the number of channels of a standard ResNet-34. Each residual block consists of convolution layers, batch

Table 1. Proposed model architecture using MCSAE (D: dimension of input feature, L: length of input feature, N: number of speakers, SE: speaker embedding).

Layer	Modified ResNet-34	Output Size	Embedding Size
conv-1	7×7, 32, stride 1	$D\times L\times32$	-
pooling-1	avg. pooling	-	1×32, ($\boldsymbol{P}_1$)
res-1	$\begin{bmatrix}3\times3, & 32\\ 3\times3, & 32\end{bmatrix}\times3$	$D\times L\times32$	-
pooling-2	avg. pooling	-	1×32, ($\boldsymbol{P}_2$)
mcsae-1	MCSAE	-	32×32, ($\boldsymbol{Z}_1$)
res-2	$\begin{bmatrix}3\times3, & 64\\ 3\times3, & 64\end{bmatrix}\times4$	$D/2\times L/2\times64$	-
pooling-3	avg. pooling	-	1×64, ($\boldsymbol{P}_3$)
mcsae-2	MCSAE	-	32×64, ($\boldsymbol{Z}_2$)
res-3	$\begin{bmatrix}3\times3, & 128\\ 3\times3, & 128\end{bmatrix}\times6$	$D/4\times L/4\times128$	-
pooling-4	avg. pooling	-	1×128, ($\boldsymbol{P}_4$)
mcsae-3	MCSAE	-	64×128, ($\boldsymbol{Z}_3$)
res-4	$\begin{bmatrix}3\times3, & 256\\ 3\times3, & 256\end{bmatrix}\times3$	$D/8\times L/8\times256$	-
pooling-5	avg. pooling	-	1×256, ($\boldsymbol{P}_5$)
mcsae-4	MCSAE	-	128×256, ($\boldsymbol{Z}_4$)
matmul	-	-	1×256, ($\boldsymbol{M}$)
concat	-	-	1×512, ($\boldsymbol{C}$)
fc-1	512×512	-	-
fc-2	512×512	-	-
fc-3	512×512	-	512×1, ($\boldsymbol{SE}$)
output	$512\times N$	-	-

normalizations, and leaky ReLU activation functions (LReLU). Especially, the output features of each residual layer is encoded in the speaker embedding in order, from low-level representation information to high-level representation information.

The output features ($\boldsymbol{P}_i$,$\boldsymbol{P}_{i+1}$) of the two previous pooling layers are used as inputs to the i^{th} MCSAE. As shown below $\boldsymbol{Z}_i$, which refers to the i^{th} segment matrix of the attention matrix $\boldsymbol{M}$ is generated by applying the random masking regularization module and cross self-attention module as in Eq. (4).

$$\boldsymbol{Z}_i = MCSAE_i(\boldsymbol{P}_i,\boldsymbol{P}_{i+1}) \qquad (0 \le i \le 4). \tag{4}$$

Here, $\boldsymbol{Z}_i$ the output of each MCSAE is used to generate an attention matrix $\boldsymbol{M}$ of 1×256 size using matrix product calculation in a *matmul* layer as in Eq. (5).

$$\boldsymbol{M}= \boldsymbol{P}_1\times\boldsymbol{Z}_1\times\boldsymbol{Z}_2\times\boldsymbol{Z}_3\times\boldsymbol{Z}_4. \tag{5}$$

To match the dimension, an embedding $\boldsymbol{P}_1$ of 1×32 size extracted from the *pooling-1* layer is used for the matrix product. Using the $\boldsymbol{P}_1$ matrix allows dimensional matching without increasing the parameters.

In the *concat* layer, embedding $\boldsymbol{P}_5$ of 1×256 size extracted from the *pooling-5* layer is concatenated with attention matrix $\boldsymbol{M}$. The embedding $\boldsymbol{P}_5$ is standard embedding in ResNet without the MCSAE. As a result, an embedding $\boldsymbol{C}$ of 1×512 size is encoded as in Eq. (6).

$$\boldsymbol{C}= concat(\boldsymbol{M},\boldsymbol{P}_5). \tag{6}$$

Finally, the concatenated embedding is encoded into fully-connected layers (*fc* layer) and output layer representing the speaker classes (*output* layer). Through this process, a 512-dimensional speaker embedding is extracted.

3.2 Cross self-attention module

The MCSAE employs two main proposed modules: 1) a cross self-attention module and, 2) a random masking regularization module. They aim to encode the segment matrix $\boldsymbol{Z}_i$ that generates the attention matrix $\boldsymbol{M}$. The MCSAE is based on the scaled dot-product attention mechanism used in the reference 16. We assume that the feature $\boldsymbol{P}_i$ is a step preceding feature $\boldsymbol{P}_{i+1}$ and they are closely related to each other, which is further emphasized by the attention mechanism. Therefore, the cross self-attention module is able to train the interdependence between the feature $\boldsymbol{P}_i$ and feature $\boldsymbol{P}_{i+1}$.

As depicted in Fig. 2, the MCSAE consists of two input pairs performing cross self-attention. The first self-attention input consists of $\boldsymbol{P}_i$ (query, $\boldsymbol{Q}$), $\boldsymbol{P}_{i+1}$ (key, $\boldsymbol{K}$), and $\boldsymbol{P}_{i+1}$ (value, $\boldsymbol{V}$). After the scaled dot-product

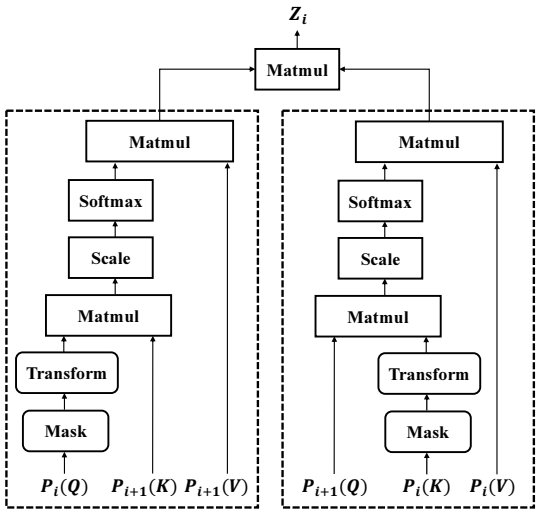

Fig. 2. Overview of the proposed MCSAE (dashed box: self-attention module, matmul: matrix multiplication).

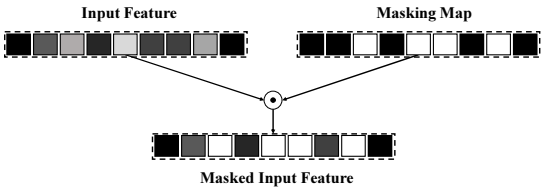

Fig. 3. Overview of the proposed random masking regularization module.

operation between $\boldsymbol{Q}$ and $\boldsymbol{K}$, self-attention is performed to the target $\boldsymbol{V}$ as in Eq. (7) (so, $\boldsymbol{P}_{i+1}$ is the attention target).

$$attention(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V}) = softmax\left(\frac{\boldsymbol{Q}^T\boldsymbol{K}}{\sqrt{d_{\boldsymbol{K}}}}\right)\boldsymbol{V}^T. \quad (7)$$

Before the scaled dot-product operation, a random masking regularization module is applied to feature $\boldsymbol{P}_i$ as shown in Fig. 3. Then, a transform layer is applied to masked $\boldsymbol{P}_i$. Here, an input feature $\boldsymbol{P}_i = [\boldsymbol{p}_1, \boldsymbol{p}_2, \cdots, \boldsymbol{p}_c, \cdots, \boldsymbol{p}_C]$ of length L ($\boldsymbol{p}_c \in \mathbb{R}^1$) is fed into the transform layer to obtain $\boldsymbol{H} = [\boldsymbol{h}_1, \boldsymbol{h}_2, \cdots, \boldsymbol{h}_c, \cdots, \boldsymbol{h}_C]$ $(\boldsymbol{h}_c \in \mathbb{R}^1)$ using LReLU activation function with slope λ as in Eq. (8).

$$\boldsymbol{h}_c = max\ \{\lambda(\boldsymbol{W}\boldsymbol{p}_c + \boldsymbol{b}), (\boldsymbol{W}\boldsymbol{p}_c + \boldsymbol{b})\}. \quad (8)$$

Next, scaling to the value of $\sqrt{d_{\boldsymbol{K}}}$ ($d_{\boldsymbol{K}}$ is the dimension of $\boldsymbol{K}$) is performed and normalization is applied using the softmax function. The computed matrix is multiplied by $\boldsymbol{V}$ and self-attention is finally conducted.

Conversely, the second self-attention input consists of $\boldsymbol{P}_{i+1}$ ($\boldsymbol{Q}$), $\boldsymbol{P}_i$ ($\boldsymbol{K}$), and $\boldsymbol{P}_i$ ($\boldsymbol{V}$). As $\boldsymbol{P}_i$ is the attention target, the scaled dot attention mechanism is performed the same as earlier. The matrix $\boldsymbol{Z}_i$ is encoded using matrix multiplication for the output of the masked cross self-attention as Eq. (9).

$$\boldsymbol{Z}_i = attention_{1^{st}}(\boldsymbol{P}_i, \boldsymbol{P}_{i+1}, \boldsymbol{P}_{i+1}) \times attention_{2^{nd}}(\boldsymbol{P}_{i+1}, \boldsymbol{P}_i, \boldsymbol{P}_i)^T. \quad (9)$$

3.3 Random masking regularization module

A random masking regularization module is applied for MCSAE, as depicted in Fig. 3. It is inspired by the Reference [19]. It aims to prevent overfitting problem in attention process of the MCSAE layer. The masking map consists of random integers, [0 or 1], according to the value of the adaptive scaling factor (default value is 0.5), which determines the range of masking that is updated by training. As the scaling factor value increases, the range of the masking widens. Then, masking is performed to input the feature and element-wise multiplication. The masked value was filled with zero.

IV. Experiments

4.1 Dataset setup

In this study, we trained the proposed model using the VoxCeleb2 dataset,[26] which contained over 1 million utterances from 5,994 celebrities. These are large-scale text-independent SV datasets collected from YouTube. We evaluated the proposed methods using the VoxCeleb1 evaluation dataset containing 40 speakers and 37,220 pairs of official test protocol.[27]

4.2 Experimental setup

The input feature vectors were 64-dimensional log Mel-filterbank energies of width 25 ms and shift size 10 ms, which were mean-variance normalized over a sliding window of up to 3 s. For each training step, 12 s interval was extracted from each utterance using cropping or padding. In training, we also used the preprocessing method to perform time and frequency masking on input features.

For parameters training, we used the standard stochastic gradient descent optimizer with a momentum of 0.9, a standard cross-entropy loss function, and a weight decay of 0.0001 at an initial learning rate of 0.1, reduced by a 0.1 decay factor on the plateau. Early stopping in 200 epochs was performed with 96 mini-batch size. The initial adaptive scaling factor was 0.5 in the random masking regularization.

Our proposed model was implemented in an end-to-end manner using PyTorch.[28] It does not used additional methods after extracting the speaker embedding such as the References [10], [14]. From the trained model, we extracted a speaker embedding and evaluated it using cosine similarity metrics: equal error rate (EER, %) performance.

4.3 Experimental results

We experimented with the proposed model using two types of comparisons. The first describes comparisons with previous self-attentive encoding in Table 2. The second describes comparisons with various state-of-the-art encodings in Table 3.

Table 2 shows the results according to the modifications of ResNet-34 up to the proposed MCSAE. We applied GAP and SAP methods to ResNet-34. In this case, 256-dimensional speaker embedding was extracted in the last residual layer. Based on MLA, the SAP was performed on the output features of each residual layer (MLA-SAP). Next, the proposed MCSAE was tested. The results showed that the proposed MCSAE performed better than the previous self-attentive encodings.

Table 2. Experimental results compared with previous encodings including SAP (Dim: dimension of speaker embedding).

Model	Encoding	Dim	EER
ResNet-34	GAP	256	4.57
	SAP	256	4.24
	MLA-SAP	512	3.49
	MCSAE (proposed)	512	2.63

Table 3. Experimental results compared with state-of-the-arts methods (*These models used VoxCeleb1 training dataset, which is smaller than the VoxCeleb2 dataset).

Model	Encoding	Dim	EER
ResNet-34[9]*	SAP	128	5.51
VGG[25]*	MHAP	512	4.00
ResNet-34[26]	TAP	512	5.04
ResNet-50[26]	TAP	512	4.19
Thin-ResNet-34[11]	NetVLAD	512	3.57
Thin-ResNet-34[11]	GhostVLAD	512	3.22
ResNet-34l*	MCSAE (proposed)	512	4.18
ResNet-34	MCSAE (propsoed)	512	2.63

Table 3 shows the results of the comparison with the state-of-the-art encodings. Here, we focused on speaker embedding encodings using a CNN-based model with the softmax loss function. These models were proposed for using various approaches such as TAP,[26] NetVLAD,[11] and GhostVLAD.[11] In addition, SAP-derived encodings were compared such as MHAP,[25] SAP.[9] The results showed that the proposed MCSAE was comparable to various state-of-the-art methods.

V. Conclusions

In this study, we proposed a new SAP-derived method for speaker embedding encoding called MCSAE. The model was focused on training both high-level and low-level layers in the ResNet architecture, in order to encode a more informative speaker embedding. In the

MCSAE, the cross self-attention module improved the concentration of the speaker information by training the interdependence among the features of each residual layer. A random masking regularization module prevented overfitting in the attention process of the MCSAE. The experimental results using the VoxCeleb1 evaluation dataset showed that the proposed MCSAE improved performance when compared with previous self-attentive encoding and state-of-the-art methods.

Acknowledgements

This work was supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIT) (No.2017-0-01772, Development of QA systems for Video Story Understanding to pass the Video Turing Test)

References

1. D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker verification using adapted Gaussian mixture models," Digital Signal Processing, **10**, 19-41 (2000).
2. W. M. Campbell, D. E. Sturim, D. A. Reynolds, and A. Solomonoff, "SVM based speaker verification using a GMM supervector kernel and NAP variability compensation," Proc. ICASSP. 97-100 (2006).
3. P. Kenny, G. Boulianne, P. Ouellet, and P. Dumouche, "Factor analysis simplified speaker verification applications," Proc. ICASSP. 637-640 (2005).
4. N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," IEEE Trans. on Audio, Speech, and Language Processing, **19**, 788-798 (2011).
5. D. Garcia-Romero and C. Y. Espy-Wilson, "Analysis of i-vector length normalization in speaker recognition systems," Proc. Interspeech, 249-252 (2011).
6. E. Variani, X. Lei, E. McDermott, I. Lopez-Moreno, and J. Gonzalez-Dominguez, "Deep neural networks for small footprint text-dependent speaker verification," Proc. ICASSP. 4052-4056 (2014).
7. D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust DNN embeddings for speaker recognition," Proc. ICASSP. 5329-5333 (2018).
8. K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," Proc. CVPR. 770-778 (2016).
9. W. Cai, J. Chen, and M. Li, "Exploring the encoding layer and loss function in end-to-end speaker and language recognition system," Proc. Odyssey, 74-81 (2018).
10. W. Cai, J. Chen, and M. Li, "Analysis of length normalization in end-to-end speaker verification system," Proc. Interspeech, 3618-3622 (2018).
11. W. Xie, A. Nagrani, J. S. Chung, and A. Zisserman, "Utterance-level aggregation for speaker recognition in the wild," Proc. ICASSP. 5791-5795 (2019).
12. I. Kim, K. Kim, J. Kim, and C. Choi, "Deep representation using orthogonal decomposition and recombination for speaker verification," Proc. ICASSP. 6126-6130 (2019).
13. Y. Jung, Y. Kim, H. Lim, Y. Choi, and H. Kim, "Spatial pyramid encoding with convex length normalization for text-independent speaker verification," Proc. Interspeech, 4030-4034 (2019).
14. S. Seo, D. J. Rim, M. Lim, D. Lee, H. Park, J. Oh, C. Kim, and J. Kim, "Shortcut connections based deep speaker embeddings for end-to-end speaker verification system," Proc. Interspeech, 2928-2932 (2019).
15. F. Wang, M. Jiang, C. Qian, S. Yang, C. Li, H. Zhang, X. Wang, and X. Tang, "Residual attention network for image classification," Proc. CVPR. 3156-3164 (2017).
16. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," Proc. NeurIPS. 5998-6008 (2017).
17. Z. Lin, M. Feng, C. N. Santos, M. Yu, B. Xiang, B. Zhou, and Y. Bengio, "A structured self-attentive sentence embedding," Proc. ICLR (2017).
18. K. Lee, X. Chen, G. Hua, H. Hu, and X. He, "Stacked cross attention for image-text matching," Proc. ECCV. 201-216 (2018).
19. L. Bao, B. Ma, H. Chang, and X. Chen, "Masked graph attention network for person re-identification," Proc. CVPR. (2019).
20. S. Zhang, Z. Chen, Y. Zhao, J. Li, and Y. Gong, "End-to-end attention based text-dependent speaker verification," Proc. SLT. 171-178 (2016).
21. G. Bhattacharya, J. Alam, and P. Kenny, "Deep speaker embeddings for short-duration speaker verification," Proc. Interspeech, 1517-1521 (2017).
22. F. R. Chowdhury, Q. Wang, I. L. Moreno, and L. Wan, "Attention-based models for text-dependent speaker verification," Proc. ICASSP. 5359-5363 (2018).
23. K. Okabe, T. Koshinaka, and K. Shinoda, "Attentive statistics pooling for deep speaker embedding," Proc. Interspeech, 2252-2256 (2018).

24. Y. Zhu, T. Ko, D. Snyder, B. Mak, and D. Povey, “Self-attentive speaker embeddings for text-independent speaker verification,” Proc. Interspeech, 3573-3577 (2018).
25. M. India, P. Safari, and J. Hernando, “Self multi-head attention for speaker recognition,” Proc. Interspeech, 4305-4309 (2019).
26. J. S. Chung, A. Nagrani, and A. Zisserman, “VoxCeleb2: Deep speaker recognition,” Proc. Interspeech, 1086-1090 (2018).
27. A. Nagrani, J. S. Chung, and A. Zisserman, “VoxCeleb: A large-scale speaker identification dataset,” Proc. Interspeech, 2616-2620 (2017).
28. A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, “Pytorch: An imperative style, high-performance deep learning library,” Proc. NeurIPS. 8024-8035 (2019).

Profile

▸ Soonshin Seo (서순신)

Soonshin Seo received his B.A. degree in Linguistics and B.E. degree in Computer Science and Engineering from Hankuk University of Foreign Studies in 2018. He is currently pursuing a Ph.D. degree in Computer Science and Engineering at Sogang University. His research interests include speaker recognition and spoken multimedia content search.

▸ Ji-Hwan Kim (김지환)

Ji-Hwan Kim received the B.E. and M.E. degrees in Computer Science from KAIST (Korea Advanced Institute of Science and Technology) in 1996 and 1998 respectively and Ph.D. degree in Engineering from the University of Cambridge in 2001. From 2001 to 2007, he was a chief research engineer and a senior research engineer in LG Electronics Institute of Technology where he was engaged in development of speech recognizer for mobile devices. In 2004, he was a visiting scientist in MIT Media Lab. Since 2007, he has been a faculty member in the Department of Computer Science and Engineering, Sogang University. Currently he is a full professor. His research interests include spoken multimedia content search, speech recognition for embedded systems and dialogue understanding.