Neural Networks 142 (2021) 329–339

Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet

2021 Special Issue

H-VECTORS: Improving the robustness in utterance-level speaker embeddings using a hierarchical attention model

Yanpei Shi *, Qiang Huang, Thomas Hain

Speech and Hearing Research Group, Department of Computer Science, University of Sheffield, UK

ARTICLE INFO

Article history:
Available online 25 May 2021

Keywords:
Speaker embeddings
Hierarchical attention
Speaker identification
Speaker verification
Attention mechanism

ABSTRACT

In this paper, a hierarchical attention network is proposed to generate robust utterance-level embeddings (H-vectors) for speaker identification and verification. Since different parts of an utterance may have different contributions to speaker identities, the use of hierarchical structure aims to learn speaker related information locally and globally. In the proposed approach, frame-level encoder and attention are applied on segments of an input utterance and generate individual segment vectors. Then, segment level attention is applied on the segment vectors to construct an utterance representation. To evaluate the quality of the learned utterance-level speaker embeddings on speaker identification and verification, the proposed approach is tested on several benchmark datasets, such as the NIST SRE2008 Part1, the Switchboard Cellular (Part1), the CallHome American English Speech ,the Voxceleb1 and Voxceleb2 datasets. In comparison with some strong baselines, the obtained results show that the use of H-vectors can achieve better identification and verification performances in various acoustic conditions.

1. Introduction

The goal of speaker recognition is to recognize a speaker from the characteristics of voices (Bai, Zhang, & Chen, 0000; Poddar, Sahidullah, & Saha, 2017). Representing the speaker properties into low dimensional feature space is beneficial for many downstream tasks, and such compact representations used to distinguish speakers (speaker embedding) have been an attractive topic and is widely used in some studies, such as speaker identification (Park, Cho, Park, Kim, & Park, 2018), verification (Le & Odobez, 2018; Novoselov, Shulipa, Kremnev, Kozlov, & Shchemelinin, 2018; Snyder, Garcia-Romero, Povey, & Khudanpur, 2017), detection (McLaren, Castan, Nandwana, Ferrer, & Yilmaz, 2018), segmentation (Garcia-Romero, Snyder, Sell, Povey, & McCree, 2017; Wang, Downey, Wan, Mansfield and Moreno, 2018), and speaker dependent speech enhancement (Chuang, Wang, Hung, Tsao, & Fang, 2019; Gao et al., 2015).

Traditionally, GMM-UBM (Dehak, Kenny, Dehak, Dumouchel, & Ouellet, 2010) based I-vectors played an important role in speaker embedding generation. With the rapid growth of deep learning techniques, previous works (rahman Chowdhury, Wang, Moreno, & Wan, 2018; Snyder, Garcia-Romero, Sell, Povey, & Khudanpur, 2018) used deep neural networks such as time delay neural networks (TDNNs) (Peddinti, Povey, & Khudanpur, 2015) and convolutional neural networks (CNNs) (Kalchbrenner, Grefenstette, & Blunsom, 2014; Zhang, Koishida and Hansen, 2018) to extract speaker embeddings. Variani, et al. developed the d-vector which uses multiple fully-connected neural network layers (Variani, Lei, McDermott, Moreno, & Gonzalez-Dominguez, 2014). In Snyder et al. (2018), Snyder, et al. proposed X-vectors, which consists of a TDNN structure that can model relationships in wide temporal contexts and computes speaker embeddings from variable length acoustic segments.

However, different parts of an utterance may have different contributions to speaker identities. How to highlight the importance of different parts of an input utterance is underdeveloped. Zhu, Ko, Snyder, Mak, and Povey (2018), for example, proposed an attentive X-vector architecture that added a global self-attention layer within the basic X-vector architecture. The attention mechanism is located prior to the statistics pooling operation. The attention mechanism computes weights for each temporal frame, and the weight vector is multiplied with the original feature map. In the output feature sequence, each of the temporal frames is assigned a weight number that indicates the importance of that frame to the target speaker identities. The results show the attentive X-vector model out-performs the original X- vector model. Both Okabe, Koshinaka, and Shinoda (2018) and Wang, Okabe, Lee, Yamamoto and Koshinaka (2018) proposed similar architectures that compute weights on different positions of the input frames. Both of the works demonstrated that attention mechanism performs better than the X-vector model in speaker recognition.

* Corresponding author.
E-mail addresses: YShi30@sheffield.ac.uk (Y. Shi), qiang.huang@sheffield.ac.uk (Q. Huang), t.hain@sheffield.ac.uk (T. Hain).

https://doi.org/10.1016/j.neunet.2021.05.024
0893-6080/

The attention mechanism highlights the most relevant part to the training target that can improve the performance of the model for different purposes (rahman Chowdhury et al., 2018). This property allowed for noise reduction methods to be developed for both image and speech signals, whereby the corrupted features were allocated lower weights to ensure that the model focuses on the clean features. In this way, excess noise can be reduced and the robustness of the model can be improved (rahman Chowdhury et al., 2018; Wang, Okabe et al., 2018). The attention mechanism used in the speaker recognition model was a self-attention layer that was built into the speaker recognition model. Similar to the self-attention mechanisms used in other domains, the attention mechanism in speaker recognition models, such as attentive X-vector and ResNet, can highlight the most relevant features in terms of the speaker identities and discard the irrelevant ones (Okabe et al., 2018; Wang, Okabe et al., 2018).

However, the attention mechanism described above has two potential problems. Firstly, the self-attention layer computes the global attention weights for each frame of the sequence. Longer sequences of, say, three seconds, may contain multiple relevant features that are important to the target speaker and, since the softmax function based global attention can only highlight some of the important features, the model is likely to lose some significant information. This is due to the fact that global attention computes the importance weight for each frame in the whole sequence. The softmax function constrains all of the attention values that can be summed to one (to simulate the probability procedure. As the sequence becomes longer, the importance of each frame is diluted (Okabe et al., 2018; Wang, Okabe et al., 2018). For example, when there are two significant features in the sequence, one of the features is captured by the attention mechanism and a high weight value is assigned (e.g. larger than 0.5). The remainder of the sequence can only share the remainder of the weighting (e.g. less than 0.5), so the second significant feature will be incorrectly weighted. This phenomenon is shown and discussed in Section 5 and Fig. 2, using the experimental results.

The second problem is that the global self-attention only captures global features, but pays insufficient attention to local features due to the computation process discussed above. In noisy conditions, as indicated by Le Prell and Clavier (2017), different types of noises (including fluctuating and steady noise) can affect the speech signal locally. For example, if one speaker-related feature (present in some frames) is distorted in some segment (or region) of the utterance, the global attention mechanism cannot capture it. For example, in Fig. 2, Section 5, the global attention mechanism cannot capture the important feature in the first segment.

In order to address the two issues discussed above, the key is to develop a new neural network architecture that can capture both local and global features in one framework. The attention mechanism needs to be used in both local and global scenarios, and this is something that can be achieved through the use of hierarchical structures like the document classification approach proposed by Yang et al. (2016). In this approach, the network firstly uses multiple word level encoders, each one of which captures the local features between words in each sentence and the attention mechanism is used to assign weights for each word within each sentence. Each sentence is then summarized in a single sentence vector. At a higher level, the generated sentence vectors are then inputted into a sentence level encoder which focuses on the global information between each sentence and the attention mechanism was used to allocate weights between each of the sentence vectors. The sentence level encoder then compresses different sentence vectors to generate a document vector, which is then used for the final prediction.

The key attribute of this approach is that the hierarchical attention structure captures the local and global information at two levels. It gristly measures the importance of each word in one sentence, then measures the importance of each sentence in one document, recognizing the fact that the importance of the same word may be different in different sentences (Yang et al., 2016).

In this paper, a hierarchical attention network is proposed, inspired by the work of Yang et al. (2016) described above, in which the utterance is viewed as a document, the segments are sentences and the frames are viewed as the words. The proposed hierarchical attention network captures the local and global speaker-related features by using the frame-level and segment-level encoders. As discussed above, some features may be corrupted in some segments but become cleaner in others.

The hierarchical attention network splits the input signal into different segments. The frame-level encoder with attention computes the attention weights between each frame within the segment. Then, the segment-level encoder measures the importance between each segment and generates the utterance vector for the final prediction of the speaker identities.

In the previous published paper: H-vectors: Utterance-level speaker embedding using a hierarchical attention model (Shi, Huang, & Hain, 2020), the hierarchical attention network is proposed and experiments were conducted on the SRE08, SWBC and CHE datasets. In this paper, besides the effectiveness of the proposed approach, its robustness against noise on several benchmark datasets, such as Voxceleb1 and Voxceleb2, will be also evaluated. Moreover, in this paper, a sliding window instead of a static window is employed to avoid missing possibly useful information for the related speaker tasks. The effectiveness of these two types of windows will be discussed in the following sections.

The rest of the paper is organized as follows: Section 2 discussed the related works, including recent works on speaker recognition and attention mechanisms. Section 3 presents the architecture of our approach. Section 4 depicts the used data, experimental setup, and the baselines to be compared. The obtained results are shown in Section 5, and a conclusion is finally drawn in Section 6.

2. Related works

The generation of speaker embeddings is a long-established task. To extract a general speaker representation, Dehak et al. (2010) defined a "total variability space" containing the speaker and channel variabilities simultaneously, and then extracted the speaker factors by decomposing feature space into subspace corresponding to sound factors including speaker and channel effects. With the rapid development of deep learning technologies, some architectures using deep neural networks (DNN) have been developed for general speaker representation (Snyder et al., 2018; Variani et al., 2014).

In Variani et al. (2014), Variani et al. introduced the *d*-vector approach using the LSTM and averaging over the activations of the last hidden layer for all frame-level features. Snyder et al. (2018) used a five-layer DNN with taking into account a small temporal context and statistics pooling. In Chung, Nagrani, and Zisserman (2018) and Xie, Nagrani, Chung, and Zisserman (2019), Chung et al. and Xie et al. applied ResNet architecture, such as ResNet-34 and thin-ResNet-34 into speaker verification.

Recently, attention mechanism is widely used in speaker embedding generation, as attention mechanism allows the model to pay attention on different parts of the input and highlight the most important part. For speaker recognition, there are some previous studies (Okabe et al., 2018; rahman Chowdhury et al., 2018; Wang, Okabe et al., 2018; Zhu et al., 2018) using attention mechanism. Wang, et al. used an attentive X-vector where a self-attention layer was added before a statistics pooling layer to

weight each frame (Okabe et al., 2018; Wang, Okabe et al., 2018; Zhu et al., 2018). Rahman, et al. jointly used attention model and K-max pooling to selects the most relevant features (rahman Chowdhury et al., 2018). Zhang, Koishida et al. (2018) used triplet loss combined with a very deep convolutional neural network to learn high quality speaker embeddings with small intra-class distances.

In addition to speaker recognition, the attention model has also been widely used in natural language processing (Bahdanau, Cho, & Bengio, 0000; Huang et al., 2016; Luong, Pham, & Manning, 0000; Yang et al., 2016), speech recognition (Chorowski, Bahdanau, Serdyuk, Cho, & Bengio, 2015; Mirsamadi, Barsoum, & Zhang, 2017; Moritz, Hori, & Le Roux, 2019; Zhang, Du, Wang, Zhang and Tu, 2018), and computer vision (Li, Tang, Deng, Zhang, & Tian, 2017; Mejjati, Richardt, Tompkin, Cosker, & Kim, 2018; Oktay et al., 0000; Wang et al., 2017; Woo, Park, Lee, & So Kweon, 2018; Xu et al., 2015). In Bahdanau et al. (0000), Bahdanau, et al. designed an attention model to allow the each time step of decoder to pay attention to different parts of the input sentence. Xu et al. used an attention model in a similar way to design an encoder decoder network for image caption (Xu et al., 2015). In Moritz et al. (2019), Moritz, et al. combined CTC (connectionist temporal classification) and attention model to improve the performance of end to end speech recognition. In Chorowski et al. (2015), Mirsamadi et al. (2017) and Zhang, Du et al. (2018), different attention models were also designed for speech emotion recognition and phoneme recognition, respectively. To further improve the robustness of the attention model, some previous studies used two attention models within one framework. Luong, et al. used global attention and local attention, where global attention attends to the whole input sentence and local attention only looks at a part of the input sentence (Luong et al., 0000). Li, et al. applied global and local attention in image processing to further improve the performance (Li et al., 2017). Woo, et al. used spatial attention and channel attention to extract salient features from input data (Woo et al., 2018).

3. Model architecture

Fig. 1 shows the architecture of the proposed hierarchical attention network. The network consists of several parts: a frame-level encoder and attention layer, a segment-level encoder and attention layer, and two fully connected layers. Given input acoustic frame vectors, the input sequence is split into several segments. The frame-level encoder and attention layers firstly compress each segment into a segment vector. Then, the segment-level encoder and attention layers generate an utterance vector from the segment vector sequence, and the following classifier is trained to perform speaker identification or verification.

3.1. Frame-level encoder and attention

For the frame-level processing, an utterance is divided into N segments: $\boldsymbol{S} = \{\boldsymbol{S}_1, \boldsymbol{S}_2, \ldots, \boldsymbol{S}_N\}$ using a sliding window with length M and step H. Each segment $\boldsymbol{S}_i \in \mathcal{R}^{M\times L} = \{\boldsymbol{x}_{i1}, \boldsymbol{x}_{i,2}, \ldots, \boldsymbol{x}_{i,M}\}$ contains M L-dimensional acoustic frame vectors $\boldsymbol{x}_{i,t} \in \mathcal{R}^{1\times L}$, where i denotes the ith segment, t denotes the tth frame, $i \in \{1, \ldots, N\}$, $t \in \{1, \ldots, M\}$.

In the frame-level encoder, a one-dimensional CNN is used on each segment, and followed by a bidirectional GRU (Chung, Gulcehre, Cho, & Bengio, 2014) in order to get information from both directions of acoustic frames and contextual information.

$$\boldsymbol{S}'_i = \mathrm{CNN}(\boldsymbol{S}_i)$$
$$\overrightarrow{\boldsymbol{h}}_i = \overrightarrow{\mathrm{GRU}}(\boldsymbol{S}'_i)$$
$$\overleftarrow{\boldsymbol{h}}_i = \overleftarrow{\mathrm{GRU}}(\boldsymbol{S}'_i)$$

The output of a frame-level encoder $\boldsymbol{h}_i = [\overrightarrow{\boldsymbol{h}}_i, \overleftarrow{\boldsymbol{h}}_i]$ contains the information of the segment $\boldsymbol{S}_i$, where $\boldsymbol{h}_i \in \mathcal{R}^{M\times E}$ and $\boldsymbol{h}_i = \{\boldsymbol{h}_{i,1}, \boldsymbol{h}_{i,2}, \ldots, \boldsymbol{h}_{i,M}\}$

In the frame-level attention layer, a two-layer MLP is first used to convert $\boldsymbol{h}_i$ into score vector $\boldsymbol{z}_i$, by which a normalized importance weight vector $\boldsymbol{\alpha}_i$ can be computed via a softmax function (Yang et al., 2016).

$$\alpha_{i,t} = \frac{\exp(z_{i,t})}{\sum_{t=0}^{M}\exp(z_{i,t})} \tag{1}$$

$$z_{i,t} = \mathrm{Relu}(\boldsymbol{h}_{i,t}\boldsymbol{W}_{i,0} + \boldsymbol{b}_{i,0})\boldsymbol{W}_{i,1} \quad , \tag{2}$$

where $z_{i,t}$ and $\alpha_{i,t}$ are a scalar score and normalized score for each time step t respectively. $\boldsymbol{W}_{i,0} \in \mathcal{R}^{E\times E}$, $\boldsymbol{b}_{i,0} \in \mathcal{R}^{1\times E}$ and $\boldsymbol{W}_{i,1} \in \mathcal{R}^{E\times 1}$ are the parameters of a two-layer MLP. These parameters are shared when processing N segments. A weighted output of frame-level encoder is computed by

$$\boldsymbol{A}_{i,t} = \alpha_{i,t}\boldsymbol{h}_{i,t} \tag{3}$$

Following Snyder et al. (2018), a statistics pooling is applied on $\boldsymbol{A}_i$ to compute its mean vector ($\boldsymbol{\mu}_i$) and std ($\boldsymbol{\sigma}_i$) vector over t. A segment vector $\boldsymbol{V}_{S_i}$ is then obtained by concatenating the two vectors:

$$\boldsymbol{V}_{S_i} = \mathrm{concatenate}(\boldsymbol{\mu}_i, \boldsymbol{\sigma}_i) \tag{4}$$

3.2. Segment level encoder and attention

For the segment-level encoder and attention, the same steps used in frame-level encoder and attention are implemented except for a bi-directional GRU layer, as the omission of the GRU layer can accelerate training when processing a large number of samples.

The output of the frame level encoder and attention is $\boldsymbol{V}_S \in \mathcal{R}^{N\times E} = \{\boldsymbol{V}_{S_1}, \boldsymbol{V}_{S_2}, \ldots, \boldsymbol{V}_{S_N}\}$. The weight vector $\boldsymbol{\alpha}^s \in \mathcal{R}^{N\times 1} = \{\alpha_1^s, \alpha_2^s, \ldots, \alpha_N^s\}$ of segment level attention can be computed as follows (Pan et al., 2019):

$$\alpha_i^s = \frac{\exp(z_i^s)}{\sum_{i=0}^{N}\exp(z_i^s)} \tag{5}$$
$$z_i^s = \mathrm{Relu}(\boldsymbol{V}_{S_i}\boldsymbol{W}_{n,0} + \boldsymbol{b}_{n,0})\boldsymbol{W}_{n,1} \quad ,$$

where z_i^s and α_i^s are a scalar score and normalized score for each segment vector $\boldsymbol{V}_{S_i}$ respectively. $\boldsymbol{W}_{n,0} \in \mathcal{R}^{E\times E}$, $\boldsymbol{b}_{n,0} \in \mathcal{R}^{1\times E}$ and $\boldsymbol{W}_{n,1} \in \mathcal{R}^{E\times 1}$ are the parameters of a two-layer MLP. A vector is generated using a statistics pooling over all weighted segments:

$$\begin{aligned}\boldsymbol{\mu}_U &= mean(\sum_i \alpha_i^s \boldsymbol{S}_i)\\ \boldsymbol{\sigma}_U &= std(\sum_i \alpha_i^s \boldsymbol{S}_i)\\ \boldsymbol{V}_U &= \mathrm{concatenate}(\boldsymbol{\mu}_U, \boldsymbol{\sigma}_U)\end{aligned} \tag{6}$$

The final speaker identity classifier is constructed using a two-layer MLP with $\boldsymbol{V}_U$ being its input. As shown in Fig. 1, the output of the first fully connected layer can be used as the final utterance embedding, represented by $\boldsymbol{E}mb_U$.

4. Experiment

4.1. Data and use

To comprehensively test the proposed approach, four datasets, NIST SRE 2008 part1 (SRE08), CallHome American English Speech (CHE), Switchboard Cellular Part 1 (SWBC), Voxceleb1 (Vox1) and Voxceleb2 (Vox2), are used in this paper to train the proposed model and evaluate utterance embedding performance.

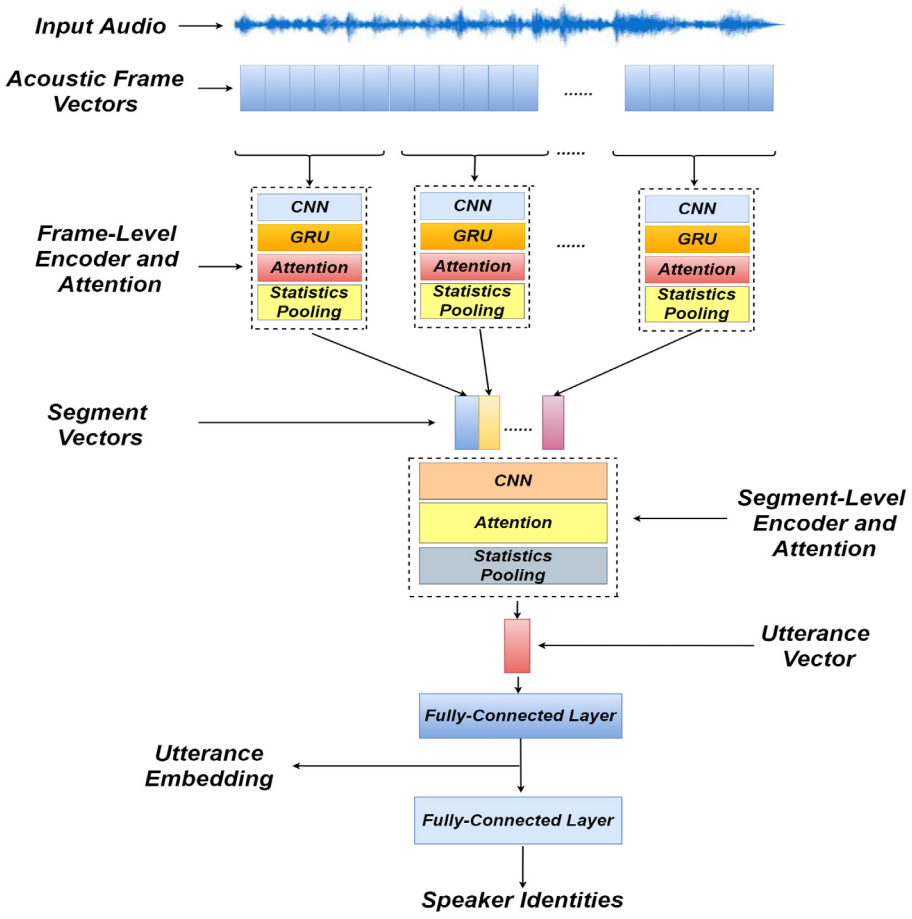

Fig. 1. The architecture of hierarchical attention network.

SRE08 indicates the 2008 NIST speaker recognition evaluation test set (Group, 2011), which contains multilingual telephone speech and English interview speech. In this work, Part1 of SRE2008, containing about 640-hour speech and 1336 distinct speakers, is selected in our experiments. The interview speech signals are approximately 3 min segmented from long conversations.

SWBC (David Graff & Walker, 2001) contains 130 h telephone speech, totally 254 speakers (129 male and 125 female) under various environment conditions (indoors, outdoors and moving vehicles). The stereo speech signals are split into two monos, and both of them are used in experiments.

CHE (Alexandra Canavan & Graff, 2001) contains 120 telephone conversations speech between native English speakers (totally 120 speakers). Among all of the calls, 90 of them are placed to various locations outside North America. In this dataset, speech from the left channel is used, as the labels of speakers in the right channels are unavailable. In our experiments, SRE08 is used to train the proposed model, by which Utterance-level embeddings can be then generated using CHE and SWBC.

The Voxceleb1 (Vox1) (Nagrani, Chung, & Zisserman, 2017) dataset is also employed as it is one of the most widely used datasets for speaker identification and verification. This dataset is extracted from Youtube videos, collected "in the wild", and has an official train–test split for both speaker identification and verification tasks. For the speaker identification task, the training set and test set contain the same number of speakers. For the speaker verification split, the test set contain 37,720 test pairs, 40 distinct speakers totally.

The Voxceleb2 (Vox2) dataset (Chung et al., 2018) is the extension of Voxceleb1, with a larger number of speakers (6112) and larger number of utterances (more than 1 mission utterances). The development set of Vox2 contains 5994 speakers while the test set contains 118 speakers. In this paper, Vox2 dataset (both development set and test set) is used for training and the test set of Vox1 is used for evaluation.

To evaluate the robustness of the proposed approach, extra noise from MUSAN dataset is used. MUSAN dataset contains three categories of noises: general noise, music and babble (Snyder, Chen, & Povey, 0000). The general noise type contains 6 h of audio, including DTMF tones, dialtones, fax machine noises etc. The music type contains 42 h of music recording from different categories. The babble type contains 60 h of speech, including read speech from public domain, hearings, committees and debates etc.

In this work, energy based VAD (Pang, 2017) is used to remove the unvoiced signals. After using VAD, each segment is viewed as an utterance. The total number of utterances of the three datasets is listed in Table 1. Each segment is further segmented into frames using a 25 ms sliding window with a 10 ms shift. All frames are converted into 20-dimensional MFCC feature vectors. Similar to Yang et al. (2016), to build a hierarchical structure, each utterance, fragment and frame vector obtained here are viewed as a document, sentence and word, respectively.

4.2. Experimental setup

In this work, both speaker identification and speaker verification tasks are conducted to evaluate the proposed model using the utterance-level embeddings. Both of the speaker identification and speaker verification experiments are split into two scenarios, each scenario uses different datasets.

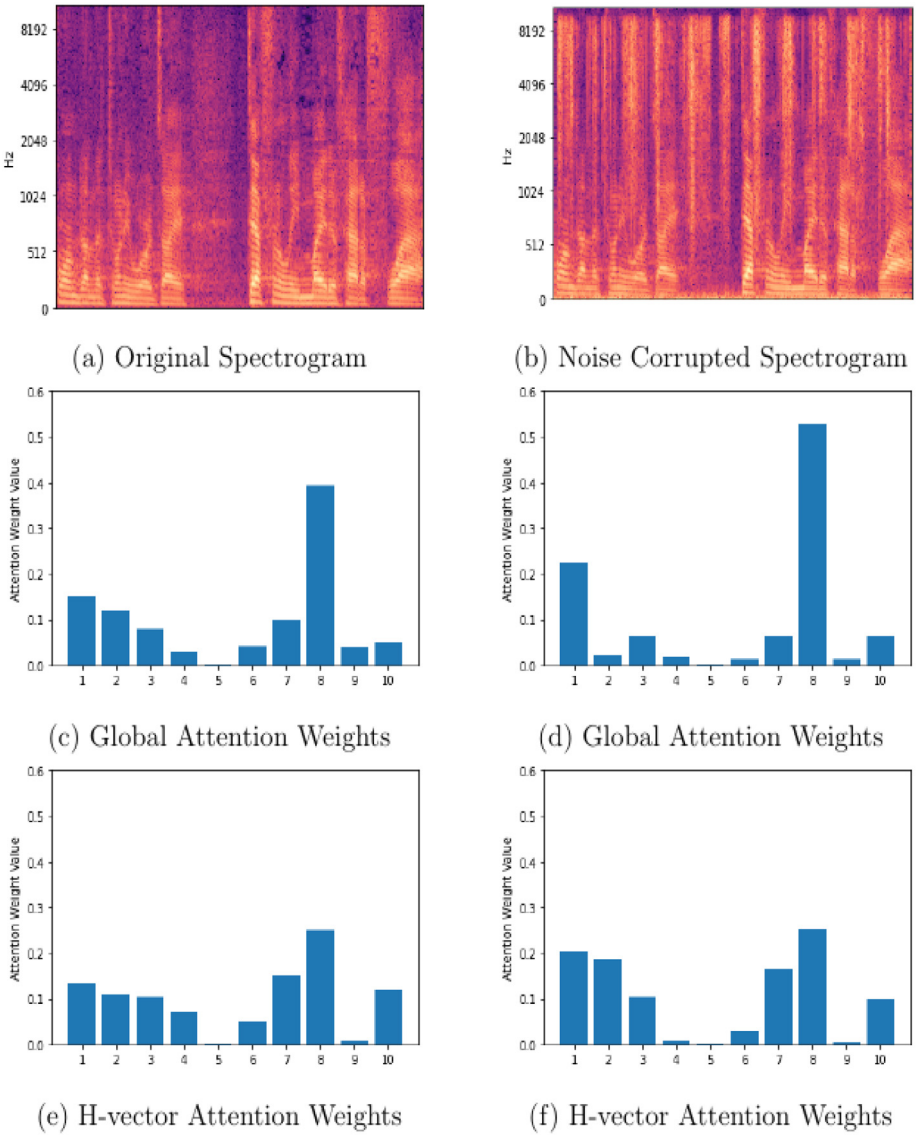

Fig. 2. The visualization of the attention weights. (a) The original spectrogram, (b) the noise corrupted spectrogram, (c) the global attention weights for the original spectrogram, (d) the global attention weights for the corrupted spectrogram, (e) the H-vector attention weights for the original spectrogram and (f) the H-vector attention weights for the corrupted spectrogram.

Table 1
The details of four speech datasets: Part1 of Sre2008 (SRE08), CallHome (CHE), Switchboard (SWBC), Voxceleb1 (Vox1) and Voxceleb2 (Vox2).

Dataset	Type	#Speaker	Size (h)	#Utterance (1 s)	#Utterance (3 s)
SRE08	Telephone+Interview	1336	640	3,528,326	1,176,453
CHE	Telephone	120	60	252,224	84,460
SWBC	Telephone	254	130	1,008,901	336,417
Vox1	Interview	1251	352	2,305,315	868,438
Vox2	Interview	6112	2442	11,408,822	3,610,387

The first scenario is to evaluate the quality of the generated utterance-level speaker embeddings. In this scenario, SRE08, CHE and SWBC datasets are used. The models are firstly trained using SRE08 dataset. Then, the trained model is used to extract utterance-level embeddings for both SEBC and CHE. The speaker identification and verification tasks are then conducted on the utterance-level embeddings

For the speaker identification task, datasets are randomly split into training and test data with 9:1 ratio. The training set and test set have the same number of speakers. For the speaker verification task, in SWBC, there are 50 speakers in the enrollment set and 120 speakers in the evaluation set, with 10 utterances for each speaker. In the CHE, there are 30 speakers in the enrollment set and 60 speakers in the evaluation set. Each speaker has 10 utterances. As a further comparison with some state-of-the-art methods, the related experiments were also conducted on the Voxceleb datasets.

The second scenario is to evaluate the robustness of the generated utterance-level speaker embedding in noise conditions. In this scenario, Voxceleb1 (Vox1) dataset is used. Vox1 dataset is recorded "in the wild", and additional noise signals are augmented.

Table 2
The architecture parameters of the proposed approach, where M denotes the segment length, N denotes the number of segments in one utterance.

Level	Model	Input	Output
Frame-level	CNN	(M,20,1)	(M,1,512)
	Bi-GRU	(M,512)	(M,1024)
	Attention	(M,1024)	(M,1024)
	Statistics Pooling	(M,1024)	(1,2048)
Segment-level	CNN	(N,2048,1)	(N,1,1500)
	Attention	(N,1500)	(N,1500)
	Statistics Pooling	(N,1500)	(1,3000)
Utterance-level	DNN (512)	(1,3000)	(1,512)
	DNN (512)	(1,512)	(1,512)

For both speaker identification and speaker verification tasks, the training sets are augmented by mixing Voxceleb1 data with noise signals from MUSAN dataset at a random SNR level (0, 5, 10, 15 and 20 dB). The test utterances are mixed with a certain type of noise with one of the five SNR levels (0, 5, 10, 15 and 20 dB).

In speaker identification task, both training and test sets contain the same number of speakers (1251 speakers) (Nagrani et al., 2017). The training set contains 145,265 utterances and the test set contains 8251 utterances. In order to reduce possible bias, the MUSAN dataset is also split into two parts for training and test. This is to ensure that the noise signals used for training will not be reused for test.

For the speaker verification task, there contains 148,642 utterances (1211 speakers) in the VoxCeleb1 development dataset, and 4874 utterances (40 speakers) in the test dataset (Nagrani et al., 2017). There are total 37,720 test pairs. The same data configuration on the data for speaker recognition task is also set for speaker verification.

For the experiments described above, both of the window size (M) and step size (H) of the proposed hierarchical attention network are fixed at 30 frames, which means there is no overlap between each segment. There is also an extra experiment to test the effectiveness when changing the window size and step size.

4.2.1. Baselines

In the experiments, some baselines, such as X-vectors (Snyder et al., 2018), attentive X-vectors (Wang, Okabe et al., 2018; Zhu et al., 2018) and ResNet (Chung et al., 2018; Xie et al., 2019) were built up for comparisons.

The first baseline (“X-Vectors”) is based on a TDNN architecture (Snyder et al., 2018). It is now widely used for speaker recognition and is effective in speaker embedding extraction. It contains a five-layer TDNN based frame-level feature extractor, each layer operating on certain time steps. A statistics pooling operation is applied on the output of the frame-level feature extractor to summarize the output sequence into a vector. Then, a DNN based segment-level feature extractor is used to generate the final speaker embedding.

The second baseline (“Attentive X-Vectors”) is made by combining a global attention mechanism with X-vectors (Okabe et al., 2018; Wang, Okabe et al., 2018; Zhu et al., 2018). In addition to the frame-level feature extractor, statistics pooling operation and the segment-level feature extractor, the Attentive X-vectors use a global attention mechanism on the output of the frame-level feature extractor before the statistics pooling operation. The attention mechanism used in Attentive X-vectors directly compute weights on each frame, which is different from the proposed approach.

The baseline of ResNet contains different variations of the ResNet architecture, such as the ResNet-34 (Chung et al., 2018) and thin ResNet-34 (Xie et al., 2019). As many works published state-of-the-art results using ResNet architecture on the Voxceleb2 dataset. As a result, the ResNet baseline is used for comparison of the proposed approach and the published state-of-the-art results.

4.2.2. Evaluation metric

In this work, prediction accuracy and the equal error rate (EER) are used as the evaluation metrics for speaker identification (Ge, Iyer, Cheluvaraja, Sundaram, & Ganapathiraju, 2017) and speaker verification (Cheng & Wang, 2004), respectively.

The models are trained by AM-softmax (m is set to 0.35, s is set to 40) (Wang, Cheng, Liu and Liu, 2018) loss function instead of the normal softmax function to achieve a better performance. AM-Softmax (additive margin softmax) aims to learn large inter-class distance and small intra-class distance for the obtained embeddings (Wang, Cheng et al., 2018). Cosine similarity is used to measure the distance of the two embeddings.

Moreover, to show the quality of the learned utterance-level embeddings, t-SNE (Maaten & Hinton, 2008) is used to visualize their distributions after being projected on a 2-dimensional plane.

4.3. Implementation

Table 2 shows the configuration of the proposed architecture. It also contains batch normalization (Ioffe & Szegedy, 2015) and dropout (Srivastava, Hinton, Krizhevsky, Sutskever, & Salakhutdinov, 2014) layers, where the dropout rate is set to 0.2. Adam optimizer (Kingma & Ba, 0000) is used for all experiments with $\beta_1 = 0.95$, $\beta_2 = 0.999$, and $\epsilon = 10^{-8}$. The initial learning rate is 10^{-4}.

5. Results

Table 3 lists some state-of-the-art results tested on Voxceleb1 when the training samples are from Voxceleb1 or Voxceleb2. It can be found that the proposed H-vectors model can outperform most of the strong baselines. The reason H-vector model can reach comparable results with the ResNet based methods (e.g. ResNet-34 or ResNet-50) may be that the hierarchical structure captures the local and global features. The frame-level encoder and attention can capture local features, which is useful to learn speaker related information within a specific region of an utterance recording and reduce the possible interferences from other regions. The segment-level encoder and attention can capture global features, this means the contributions from different regions of an utterance will be balanced.

In order to show how the attention mechanism works, Fig. 2 shows the visualization of the attention weights. Fig. 2(a) is the spectrogram of a 3s utterance randomly selected from the Voxceleb1 dataset. Fig. 2(b) shows the noise corrupted spectrogram (with 0 dB). For a better visualization, here demonstrate spectrograms, instead of MFCCs, of utterance recordings. Figs. 2 (c) and (d) show the attention weights obtained by using the attentive X-vector (global attention) on the original utterance and the noise corrupted utterance respectively. Figs. 2 (e) and (f) show the attention weights obtained by using the H-vector in the same conditions. Note that the number of the attention weights in the attentive X-vector is 300 (there are 300 frames in the input data) and the number of the segment-level attention weights in H-vector is 10 (10 segment vectors). In order to compare the attention weights, the attention weights of the attentive X-vector are divided into 10 groups.

Although the weight distributions displayed in Fig. 2(c) and (d) show that the use of both attentive X-vector and H-vector can learn the importance of features in different parts of an utterance recording, the attentive X-vector assigned a high weight value, about 0.5 to the 8th segment. This means the contribution of

Table 3
The comparison of the proposed approach with the state-of-the-art on Voxceleb1 test set.

	Model	Training set	Loss	EER %
Nagrani et al. (2017)	VGG-M	Voxceleb1	Softmax	10.2
Nagrani et al. (2017)	VGG-M	Voxceleb1	Softmax+Contrastive	7.8
Shon, Tang, and Glass (2019)	CNN+TDNN	Voxceleb1	Softmax	6.79
Cai, Chen, and Li (0000)	ResNet-34	Voxceleb1	A-Softmax+PLDA	4.46
Okabe et al. (2018)	X-vector (TAP)	Voxceleb1	Softmax+PLDA	4.70
Okabe et al. (2018)	X-vector (SAP)	Voxceleb1	Softmax+PLDA	4.19
Okabe et al. (2018)	X-vector (ASP)	Voxceleb1	Softmax+PLDA	3.85
Hajibabaei and Dai (0000)	ResNet20	Voxceleb1	A-Softmax	4.40
Hajibabaei and Dai (0000)	Retnet-20	Voxceleb1	AM-Softmax	4.30
Ours	H-vector	Voxceleb1	AM-Softmax	4.28
Chung et al. (2018)	VGG-M	Voxceleb2	Softmax+Contrastive	5.94
Chung et al. (2018)	ResNet-34	Voxceleb2	Softmax+Contrastive	5.04
Chung et al. (2018)	ResNet-34	Voxceleb2	Softmax+Contrastive	4.83
Chung et al. (2018)	ResNet-50	Voxceleb2	Softmax+Contrastive	4.19
Chung et al. (2018)	ResNet-50	Voxceleb2	Softmax+Contrastive	4.43
Chung et al. (2018)	ResNet-50	Voxceleb2	Softmax+Contrastive	3.95
Xie et al. (2019)	Thin-ResNet-34	Voxceleb2	Softmax+TAP	10.48
Xie et al. (2019)	Thin-ResNet-34	Voxceleb2	Softmax+NetVLAD	3.57
Xie et al. (2019)	Thin-ResNet-34	Voxceleb2	AM-Softmax+NetVLAD	3.32
Xie et al. (2019)	Thin-ResNet-34	Voxceleb2	Softmax+GhostVLAD	3.22
Xie et al. (2019)	Thin-ResNet-34	Voxceleb2	AM-Softmax+GhostVLAD	3.23
Nagrani, Chung, Xie, and Zisserman (2020)	Thin-ResNet-34	Voxceleb2	AM-Softmax+GhostVLAD	2.87
Ours	H-vector	Voxceleb2	AM-Softmax	3.63
Ours	H-vector	Voxceleb2	AM-Softmax+Contrastive	3.21

Table 4
Identification accuracy on the test data of SRE08 when the utterance length is 1 s or 3 s. *M* and *H* are set to 30 frames.

Utterance length	Model	Accuracy %
1 s	X-vector	90.1
	X-vector+Att	92.1
	H-vector	94.5
3 s	X-vector	95.2
	X-vector+Att	96.7
	H-vector	98.5

Table 5
Identification accuracy and equal error rate (EER) on CHE dataset when the utterance length is 1 s or 3 s. *M* and *H* are set to 30 frames. The previous published results (Pre) are also listed (Shi et al., 2020).

Utterance length	Model	Accuracy %	EER % (Pre)	EER %
1 s	X-vector	84.8	1.94	1.86
	X-vector+Att	87.5	1.61	1.53
	H-vector	89.1	1.44	1.36
3 s	X-vector	89.4	1.46	1.39
	X-vector+Att	91.0	1.21	1.18
	H-vector	92.8	1.08	1.01

the 8th segment is dominant over the rest 9 segments. This might easily cause an overestimate, and thus probably lead to an incorrect decision. As a comparison, although the H-vector model allocated the highest weight to the 8th segment, it is close to 0.3 as shown in Fig. 2(e) and (f), and other segments are also allocated a relatively reasonable attention values. It shows the H-vector model can highlight feature contributions from multiple regions of an utterance recording.

It may be that the global attention process within the attentive X-vector model may tend to favour few number of regions over others of a recording, whereas the hierarchical structure of the H-vector model is able to highlight contributions from more regions by computing the attention weights within a small segment, and then computing the attention weights over all segments.

Table 4 shows the identification accuracy on the test data of SRE08 using the proposed approach and two baselines. Two different utterance lengths, 1 s and 3 s, are used in the experiments, respectively to evaluate the performance of the models in short and long input utterances. The use of the H-vectors shows higher accuracy when using either 1-second or 3-second input length than the two baselines. When the length of input utterances is 1 s, the accuracy obtained using the H-vectors can reach 94.5%, with 4.4% improvements over X-vectors and 2.4% improvement over X-vectors+Attention, respectively. When the length of input utterances is 3 s, the accuracy obtained using the H-vectors can reach 98.5%, with about 3% improvement over X-vectors and about 2% improvements over X-vectors+Attention. The proposed approach is more robust than the two baselines when processed utterances are short. In addition, the accuracies obtained using 3-second utterances are better than those using 1-second utterances. This probably means a longer utterance may contain more information relevant to a target speaker than short ones.

To evaluate the quality of embeddings extracted using the proposed approach and its robustness on out-of-domain data, two additional datasets (SEBC and CHE) are employed in our experiments. Tables 5 and 6 show the identification accuracy and verification equal error rate when using the embeddings learned on the SWBC and the CHE dataset, respectively. In these two tables, the previous published results are also listed (Shi et al., 2020). The previous work used different post-processing techniques for the obtained embeddings: The models are trained using normal softmax function, PLDA back-end (Salmun, Opher, & Lapidot, 2016) is applied on the embeddings to reduce the dimension to 300 (Shi et al., 2020).

On the two datasets, the H-vectors consistently outperforms the two baselines whether the length of utterances is 1 s or 3 s. In CHE dataset, the H-vector approach reaches 89.1% prediction accuracy and 1.44% equal error rate, with more than 3% improvement than X-vectors and Attentive X-vectors in speaker identification task. In speaker verification task, the H-vectors also achieved 3% relative improvement than X-vectors and attentive X-vectors. Similar to the results in SRE08 dataset, the results obtained by three-second utterance length are better than that using one-second utterance length.

For most of the cases, the results obtained by this work are slightly better than that of the previous published results. The reason might be the use of AM-Softmax function in the training process.

Table 6
Identification accuracy and Equal Error Rate (EER) on SWBC dataset when the utterance length is 1 s or 3 s. *M* and *H* are set to 30 frames. The previous published results (Pre) are also listed (Shi et al., 2020).

Utterance length	Model	Accuracy %	EER % (Pre)	EER %
1 s	X-vector	78.2	2.23	2.17
	X-vector+Att	81.0	2.05	2.02
	H-vector	83.7	1.92	1.90
3 s	X-vector	81.3	2.01	1.98
	X-vector+Att	84.0	1.82	1.79
	H-vector	86.2	1.69	1.61

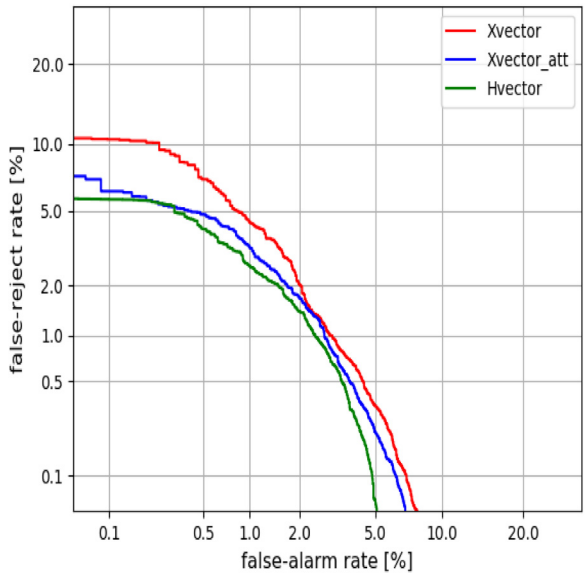

Fig. 3. The DET curve on the SEBC dataset when the segment length is 3 s.

From the results in Tables 4–6, it is obvious that the best results are obtained by SRE08 dataset. The results obtained on SWBC dataset are lower than those on the other two datasets. Since the model is trained on the SRE08 corpus, the identification performances on its test data are clearly better than those on the other two datasets. In comparison with SRE08 dataset, both CHE and SWBC could be viewed as out-of-domain dataset. There might be some mis-match between the test sets of CHE and SWBC dataset and the SRE08 dataset (used for training). Furthermore, as the SWBC dataset contains a wide range of environment conditions (indoors, outdoors and moving vehicles), the acoustic conditions are worse than SRE08 and CHE dataset. As a result, both its identification and verification performances are relatively worse than those obtained on the CHE dataset.

To further show the performance of the proposed model, Fig. 3 illustrates the detection error tradeoff (DET) curve of the three models (X-vectors, Attentive X-vectors and H-vectors) on SWBC dataset when the utterance length is 3 s. From Fig. 3, it is clear that H-vectors obtained both lower false reject rate and false alarm rate, and yield lower equal error rate. Attentive X-vectors obtained higher false reject rate and false alarm rate, but still lower than that obtained by X-vectors. This might be mainly due to the use of attention mechanism. Attentive X-vectors use global attention that allocates different weights on each frame, which could highlight the importance of different frames. However, with the combination of local and global attention, the proposed hierarchical attention could out-perform the attentive X-vectors, reaching lower false reject rate and false alarm rate.

Table 7
Identification accuracy and Equal Error Rate (EER) on Voxceleb1 dataset when the utterance length is 1 s or 3 s. *M* and *H* are set to 30 frames.

Utterance length	Model	Accuracy %	EER %
1 s	X-vector	85.8	5.75
	X-vector+Att	86.9	5.22
	H-vector	88.7	4.97
3 s	X-vector	88.2	5.13
	X-vector+Att	89.2	4.79
	H-vector	90.4	4.64

Table 8
Identification accuracy and Equal Error Rate (EER) on Voxceleb1 dataset when the window size *M* is changed from 15 to 35 frames.

Utterance length	Window size	Accuracy %	EER %
1 s	15	86.4	5.24
	20	87.3	5.01
	25	89.2	4.82
	30	88.7	4.97
	35	88.3	5.11
3 s	15	88.7	4.72
	20	89.6	4.43
	25	91.0	4.28
	30	90.4	4.64
	35	89.5	4.79

To further evaluate the quality of extracted utterance-level embeddings, t-SNE (Maaten & Hinton, 2008) is used to visualize the distribution of embeddings by projecting these high-dimensional vectors on a 2D plane. In the SWBC dataset, 10 speakers are selected and 500 three-second segments are randomly sampled for each speaker. Fig. 4(a), (b), and (c) show the distribution of selected samples of 10 speakers after using X-vectors, X-vectors+Attention, and H-vectors, respectively. Each colour represents a distinct speaker and each point represents an utterance. The black mark represents the centre point of each speaker class. Fig. 4(a) shows the distribution of the embeddings obtained by X-vectors. It is clear that, in this figure, some samples from different speakers are not well discriminated as there are overlaps between speaker classes. Due to the use of an attention mechanism in X-vectors+Attention, Fig. 4(b) shows a better sample distribution than Fig. 4(a). However, some samples of a speaker labelled by a blue colour are not well clustered. In Fig. 4(c), the embedding obtained by H-vectors performs a better separation than the baseline methods.

In the second scenario, Voxceleb1 dataset is used to evaluate the proposed approach. In this scenario, the three models (X-vectors, Attentive X-vectors and H-vectors) are trained using the official training set of Voxceleb1 for speaker identification and verification tasks. Table 7 shows the speaker identification accuracy and equal error rate on the Voxceleb1 dataset. Similar to the results in the previous three datasets, H-vectors show better performance on Voxceleb1 dataset. In speaker identification task, H-vectors achieved 88.7% accuracy in when the utterance length is 1 s and 90.4% accuracy when the utterance length is 3 s. H-vectors obtain more than 3% relatively improvement than X-vectors and Attentive X-vectors. In speaker verification task, H-vectors reach 4.97% equal error rate on one-second utterance length and 4.64% on three-second utterance length, the improvement is also significant.

In the hierarchical attention network architecture, different window size (M) and step size (H) might influence the performance. In order to evaluate the performance of the proposed H-vectors when using different window size (M) and step size (H), Tables 8 and 9 show the prediction accuracy and equal error rate on Voxceleb1 dataset when the window size changes from 15 to 35 frames, and the step size changes from 15 to 35 frames.

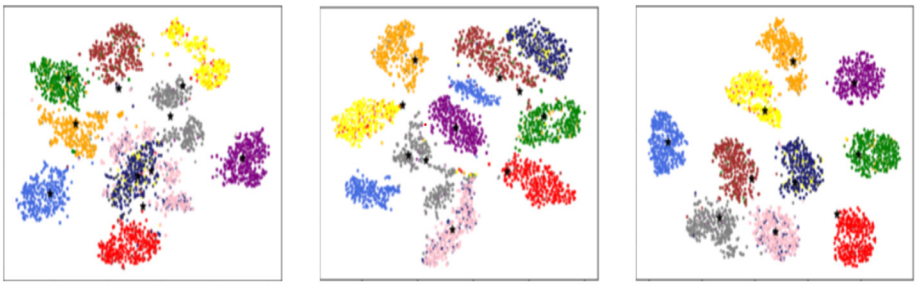

Fig. 4. Embedding visualization using t-SNE. In the SWBC dataset, 10 speakers are selected and 500 three-second segments are randomly sampled for each speaker. Each colour represents a speaker, and each point indicates an utterance. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 9
Identification accuracy and Equal Error Rate (EER) on Voxceleb1 dataset when the step size H is changed from 15 to 35 frames.

Utterance length	Step size	Accuracy %	EER %
1 s	15	87.5	4.93
	20	89.6	4.86
	25	89.4	4.92
	30	88.7	4.97
	35	87.1	5.12
3 s	15	90.1	4.61
	20	91.0	4.43
	25	90.6	4.37
	30	90.4	4.64
	35	88.3	4.90

Table 10
Speaker identification results for different noise types (Noise, Music and Babble) at different SNR (0–20 dB), and the original Voxceleb1 test set. The utterance length is 3 s. M and H are set to 30 frames.

Noise type	SNR	X-vectors	Att-X-vectors	Statistical	H-vectors
Noise	0	74.6	75.8	73.8	76.9
	5	79.5	79.4	78.7	81.3
	10	83.1	84.0	83.8	86.0
	15	85.0	86.3	85.9	87.2
	20	87.9	87.8	86.7	88.9
Music	0	68.2	70.1	66.7	72.3
	5	72.0	73.5	71.4	74.8
	10	79.4	81.0	79.5	82.9
	15	84.2	86.6	83.3	87.8
	20	86.1	88.0	85.2	89.3
Babble	0	64.1	65.2	62.1	67.9
	5	70.5	71.4	68.4	74.0
	10	77.4	77.0	76.4	78.7
	15	83.5	84.5	81.8	86.2
	20	86.6	86.9	86.0	88.1
Original		88.2	89.2	87.6	90.4

From the results, the model is more sensitive to the change of the window size. The best performance of is obtained when M is equal to 25 frames for both one or three seconds segment length. While, for the change of the step size, the best performance is obtained when H is equal to 20 frames. One possible reason is that the use of sliding window (the window size is larger than the step size) instead of static window (the window size is equal to the step size) might capture more information.

In order to evaluate the robustness of the proposed model in noise conditions, additional noises from MUSAN dataset are mixed with the utterances from the original Voxceleb1 dataset. Tables 10 and 11 show the speaker identification accuracy and speaker verification equal error rate on different noise conditions. Three noise types are used: general noise, music and speech noise. The noise level is changed from 0 dB to 20 dB. The utterance length is three seconds.

Table 11
Speaker verification results for different noise types (Noise, Music and Babble) at different SNR (0–20 dB), and the original Voxceleb1 test set. The utterance length is 3 s. M and H are set to 30 frames.

Noise type	SNR	X-vectors	Att-X-vectors	Statistical	H-vectors
Noise	0	12.26	11.32	12.82	10.92
	5	10.01	9.26	11.03	9.03
	10	8.33	7.77	8.92	7.28
	15	7.25	6.76	8.14	6.50
	20	6.91	6.02	7.48	5.95
Music	0	14.15	12.92	15.88	12.68
	5	11.03	10.04	12.20	9.83
	10	9.35	8.64	10.69	8.33
	15	8.41	8.08	9.83	7.62
	20	6.79	6.25	7.72	6.17
Babble	0	30.02	27.77	32.56	26.82
	5	16.46	15.32	18.02	14.58
	10	13.26	12.53	15.38	12.38
	15	9.10	8.31	10.47	8.14
	20	7.95	7.22	8.91	7.04
Original		5.47	5.06	5.93	4.64

From the results, the proposed H-vectors outperform the two baselines in different noise conditions. When the noise type becomes complex and the noise level becomes larger, such as babble and music noise type at 0 and 5 dB, the gap between the results of H-vectors is larger than that of the two baselines. Even if the noise type is "Babble" and the noise level is 0 dB, the proposed H-vectors model can reach 67.7% prediction accuracy, and obtain more than 5% relative improvement than X-vectors and 3% relative improvement than Attentive X-vectors.

The "Statistical" in Tables 10 and 11 represents the H-vector model that is without the attention mechanism in both frame-level and segment-level encoders. In this case, only statistical pooling operation is used to compress the sequence input a vector, without allocating weights for each frame. This is to evaluate the effectiveness of the attention mechanism. The results show that the H-vector with attention out-performs that without attention mechanism under almost all of the noise conditions. When the noise level becomes larger, the gap between H-vector with attention and that without attention becomes larger. This phenomenon shows the local and global attention mechanisms are essential for the H-vector model, they can help the model to improve the robustness.

6. Conclusion and future work

In this paper, a hierarchical attention network was proposed for utterance-level embedding extraction. Inspired by the hierarchical structure of a document made by words and sentences,

each utterance is viewed as a document, segments and frame vectors are treated as sentences and words, respectively. The use of attention mechanisms at frame and segment levels provides a way to search for the information relevant to target locally and globally, thus obtains better utterance level embeddings, including better performances on speaker identification and verification tasks, and better performances in various noise conditions.

In the future work, different attention mechanisms, such as the multi-head attention mechanism will be investigated in the hierarchical structure. Moreover, the attention mechanism in different dimensions of the input data, such as time and frequency dimensions, will be tested.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: This work was in part supported by Innovate UK Grant number 104264

Acknowledgements

This work was in part supported by Innovate UK Grant number 104264 MAUDIE.

References

Alexandra Canavan, G. Z., & Graff, David (2001). Callhome american english speech. https://catalog.ldc.upenn.edu/LDC97S42.

Bahdanau, D., Cho, K., Bengio, Y., & Neural machine translation by jointly learning to align and translate, arXiv:1409.0473.

Bai, Z., Zhang, X.-L., & Chen, J. Speaker recognition based on deep learning: An overview. arXiv preprint arXiv:2012.00931.

Cai, W., Chen, J., & Li, M. Exploring the encoding layer and loss function in end-to-end speaker and language recognition system. arXiv preprint arXiv:1804.05160.

Cheng, J.-M., & Wang, H.-C. (2004). A method of estimating the equal error rate for automatic speaker verification. In *2004 international symposium on chinese spoken language processing* (pp. 285–288). IEEE.

Chorowski, J. K., Bahdanau, D., Serdyuk, D., Cho, K., & Bengio, Y. (2015). Attention-based models for speech recognition. In *NIPS* (pp. 577–585).

Chuang, F.-K., Wang, S.-S., Hung, J.-w., Tsao, Y., & Fang, S.-H. (2019). Speaker-aware deep denoising autoencoder with embedded speaker identity for speech enhancement. In *Interspeech* (pp. 3173–3177).

Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. In *NIPS*.

Chung, J. S., Nagrani, A., & Zisserman, A. (2018). Voxceleb2: Deep speaker recognition. *Proceedings of Interspeech*, *2018*, 1086–1090.

David Graff, D. M., & Walker, Kevin (2001). Switchboard cellular part 1 audio. https://catalog.ldc.upenn.edu/LDC2001S13.

Dehak, N., Kenny, P. J., Dehak, R., Dumouchel, P., & Ouellet, P. (2010). Front-end factor analysis for speaker verification. *IEEE Transactions on Audio, Speech, and Language Processing*, *19*(4), 788–798.

Gao, T., Du, J., Xu, L., Liu, C., Dai, L.-R., & Lee, C.-H. (2015). A unified speaker-dependent speech separation and enhancement system based on deep neural networks. In *ChinaSIP*. IEEE.

Garcia-Romero, D., Snyder, D., Sell, G., Povey, D., & McCree, A. (2017). Speaker diarization using deep neural network embeddings. In *ICASSP* (pp. 4930–4934). IEEE.

Ge, Z., Iyer, A. N., Cheluvaraja, S., Sundaram, R., & Ganapathiraju, A. (2017). Neural network based speaker classification and verification systems with enhanced features. In *2017 intelligent systems conference (IntelliSys)* (pp. 1089–1094). IEEE.

N. M. I. Group (2011). 2008 nist speaker recognition evaluation training set part 1. https://catalog.ldc.upenn.edu/LDC2011S05.

Hajibabaei, M., & Dai, D. Unified hypersphere embedding for speaker recognition. arXiv preprint arXiv:1807.08312.

Huang, X., et al. (2016). Attention-based convolutional neural network for semantic relation extraction. In *Proceedings of COLING 2016, the 26th international conference on computational linguistics: Technical papers* (pp. 2526–2536).

Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML* (pp. 448–456).

Kalchbrenner, N., Grefenstette, E., & Blunsom, P. (2014). A convolutional neural network for modelling sentences. In *ACL (Volume 1: Long papers)* (pp. 655–665).

Kingma, D. P., & Ba, J. Adam: A method for stochastic optimization. CoRR abs/1412.6980.

Le, N., & Odobez, J.-M. (2018). Robust and discriminative speaker embedding via intra-class distance variance regularization. In *Interspeech* (pp. 2257–2261).

Le Prell, C. G., & Clavier, O. H. (2017). Effects of noise on speech recognition: Challenges for communication by service members. *Hearing Research*, *349*, 76–89.

Li, L., Tang, S., Deng, L., Zhang, Y., & Tian, Q. (2017). Image caption with global-local attention. In *AAAI*.

Luong, M.-T., Pham, H., & Manning, C. D. Effective approaches to attention-based neural machine translation. arXiv:1508.04025.

Maaten, L. v. d., & Hinton, G. (2008). Visualizing data using t-sne. In *JMLR* (pp. 2579–2605).

McLaren, M., Castan, D., Nandwana, M., Ferrer, L., & Yilmaz, E. (2018). How to train your speaker embeddings extractor. In *Proc. speaker odyssey, les sables d'olonne, France: ISCA* (pp. 327–334).

Mejjati, Y. A., Richardt, C., Tompkin, J., Cosker, D., & Kim, K. I. (2018). Unsupervised attention-guided image-to-image translation. In *NIPS* (pp. 3693–3703).

Mirsamadi, S., Barsoum, E., & Zhang, C. (2017). Automatic speech emotion recognition using recurrent neural networks with local attention. In *ICASSP* (pp. 2227–2231). IEEE.

Moritz, N., Hori, T., & Le Roux, J. (2019). Triggered attention for end-to-end speech recognition. In *ICASSP* (pp. 5666–5670). IEEE.

Nagrani, A., Chung, J. S., Xie, W., & Zisserman, A. (2020). Voxceleb: Large-scale speaker verification in the wild. *Computer Speech and Language*, *60*, Article 101027.

Nagrani, A., Chung, J. S., & Zisserman, A. (2017). Voxceleb: a large-scale speaker identification dataset. *Telephony*, *3*, 33–039.

Novoselov, S., Shulipa, A., Kremnev, I., Kozlov, A., & Shchemelinin, V. (2018). On deep speaker embeddings for text-independent speaker recognition. In *Proc. Odyssey 2018 the speaker and language recognition workshop* (pp. 378–385).

Okabe, K., Koshinaka, T., & Shinoda, K. (2018). Attentive statistics pooling for deep speaker embedding. In *Interspeech* (pp. 2252–2256).

Oktay, O., Schlemper, J., Folgoc, L. L., Lee, M., Heinrich, M., Misawa, K., et al. Attention u-net: Learning where to look for the pancreas. arXiv:1804.03999.

Pan, Y., Mirheidari, B., Reuber, M., Venneri, A., Blackburn, D., & Christensen, H. (2019). Automatic hierarchical attention neural network for detecting ad. In *Interspeech* (pp. 4105–4109).

Pang, J. (2017). Spectrum energy based voice activity detection. In *CCWC* (pp. 1–5).

Park, H., Cho, S., Park, K., Kim, N., & Park, J. (2018). Training utterance-level embedding networks for speaker identification and verification. In *Interspeech* (pp. 3563–3567).

Peddinti, V., Povey, D., & Khudanpur, S. (2015). A time delay neural network architecture for efficient modeling of long temporal contexts. In *ISCA*.

Poddar, A., Sahidullah, M., & Saha, G. (2017). Speaker verification with short utterances: a review of challenges, trends and opportunities. *IET Biometrics*, *7*(2), 91–101.

rahman Chowdhury, F. R., Wang, Q., Moreno, I. L., & Wan, L. (2018). Attention-based models for text-dependent speaker verification. In *ICASSP* (pp. 5359–5363). IEEE.

Salmun, I., Opher, I., & Lapidot, I. (2016). On the use of plda i-vector scoring for clustering short segments. In *Odyssey* (pp. 407–414).

Shi, Y., Huang, Q., & Hain, T. (2020). H-vectors: Utterance-level speaker embedding using a hierarchical attention model. In *ICASSP 2020-2020 IEEE international conference on acoustics, speech and signal processing (ICASSP)* (pp. 7579–7583). IEEE.

Shon, S., Tang, H., & Glass, J. (2019). Voiceid loss: Speech enhancement for speaker verification. *Proceedings of Interspeech*, *2019*, 2888–2892.

Snyder, D., Chen, G., & Povey, D. Musan: A music, speech, and noise corpus. arXiv preprint arXiv:1510.08484.

Snyder, D., Garcia-Romero, D., Povey, D., & Khudanpur, S. (2017). Deep neural network embeddings for text-independent speaker verification. In *Interspeech* (pp. 999–1003).

Snyder, D., Garcia-Romero, D., Sell, G., Povey, D., & Khudanpur, S. (2018). X-vectors: Robust dnn embeddings for speaker recognition. In *ICASSP* (pp. 5329–5333). IEEE.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. In *JMLR* (pp. 1929–1958).

Variani, E., Lei, X., McDermott, E., Moreno, I. L., & Gonzalez-Dominguez, J. (2014). Deep neural networks for small footprint text-dependent speaker verification. In *ICASSP*. IEEE.

Wang, F., Cheng, J., Liu, W., & Liu, H. (2018). Additive margin softmax for face verification. *IEEE Signal Processing Letters*, *25*(7), 926–930.

Wang, Q., Downey, C., Wan, L., Mansfield, P. A., & Moreno, I. L. (2018). Speaker diarization with lstm. In *ICASSP* (pp. 5239–5243). IEEE.

Wang, F., Jiang, M., Qian, C., Yang, S., Li, C., Zhang, H., et al. (2017). Residual attention network for image classification. In *CVPR* (pp. 3156–3164).
Wang, Q., Okabe, K., Lee, K. A., Yamamoto, H., & Koshinaka, T. (2018). Attention mechanism in speaker recognition: What does it learn in deep speaker embedding?. In *SLT* (pp. 1052–1059). IEEE.
Woo, S., Park, J., Lee, J.-Y., & So Kweon, I. (2018). Cbam: Convolutional block attention module. In *ECCV* (pp. 3–19).
Xie, W., Nagrani, A., Chung, J. S., & Zisserman, A. (2019). Utterance-level aggregation for speaker recognition in the wild. In *ICASSP 2019-2019 IEEE international conference on acoustics, speech and signal processing (ICASSP)* (pp. 5791–5795). IEEE.
Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhudinov, R., et al. (2015). Show, attend and tell: Neural image caption generation with visual attention. In *ICML* (pp. 2048–2057).
Yang, Z., Yang, D., Dyer, C., He, X., Smola, A., & Hovy, E. (2016). Hierarchical attention networks for document classification. In *NAACL* (pp. 1480–1489).
Zhang, Y., Du, J., Wang, Z., Zhang, J., & Tu, Y. (2018). Attention based fully convolutional network for speech emotion recognition. In *APSIPA ASC* (pp. 1771–1775). IEEE.
Zhang, C., Koishida, K., & Hansen, J. H. (2018). Text-independent speaker verification based on triplet convolutional neural network embeddings. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *26*(9), 1633–1644.
Zhu, Y., Ko, T., Snyder, D., Mak, B., & Povey, D. (2018). Self-attentive speaker embeddings for text-independent speaker verification. In *Interspeech* (pp. 3573–3577).