Combination of Multiple Embeddings for Speaker Retrieval

Xinmei Su\textsuperscript{1}, Qingran Zhan\textsuperscript{1}, Chenguang Hu\textsuperscript{1}, Xiang Xie\textsuperscript{1,2}

\textsuperscript{1}School of Information and Electronics, Beijing Institute of Technology, Beijing, China
\textsuperscript{2}Shenzhen Research Institute, Beijing Institute of Technology, Shenzhen, China

suxinmei2022@126.com

Abstract

Speaker retrieval (SR) is a task to select the enrolled speakers from a large amount of test utterances. Extracting speaker embeddings in retrieval tasks depends on deep neural networks in general. The Emphasized Channel Attention, Propagation and Aggregation in Time Delay Neural Network (ECAPA-TDNN), which is the state-of-the-art (SOTA) neural network in the field of speaker verification (SV), can also be used in solving SR problems. In this paper, we propose an extension of architecture based on ECAPA-TDNN that combines multiple embeddings in different layers. First, we replace the front TDNN layers in ECAPA-TDNN with multi-scale convolution layers that are adopted by multi-scale 1-D convolutional kernels. By applying multi-scale convolution, multiple scales of feature maps are extracted and multiple information is learned by the neural network. Second, skip connections in SE-Res2blocks are added to avoid overfitting. Third, a novel pooling method is employed and concatenated with the statistic attentive pooling to achieve better performances. Combination of multiple poolings can help the network get more spatial features. The proposed system obtains a relative improvement of 22.7\% comparing with the SOTA model before. A further qualitative analysis proves the results even further.

Index Terms: speaker retrieval, ECAPA-TDNN, multi-scale convolution

1. Introduction

Speaker retrieval (SR) is widely applied in daily life\cite{1}, such as the detection of speakers in complex environments with strong background noises. The speaker retrieval system is able to fully automatically screen out correct utterances from a large data pool given the true identity of each speaker. As the data pool for SR task tends to be large and complicated, the retrieval system should be able to accurately distinguish various speakers. First, it should filter out non-verbal sounds and avoid their influence. Second, the most relevant utterances from the same speaker need to be collected into a cluster. Third, the system should be able to identify each cluster group with the label of the corresponding speaker\cite{2}.

Traditional methods of extracting speaker embedding for speaker verification and retrieval are based on Gaussian mixture model (GMM)\cite{3,4}, such as i-vector\cite{5} system. However, i-vector contains both the information of speakers and channel differences. It is not only necessary to remove the channel interference in the i-vector system, but also to eliminate the channel interference by channel compensation technology. In the past few decades, the application of deep neural networks leads to breakthroughs in the field of speaker retrieval. The x-vector\cite{6} system with a Time Delay Neural Network (TDNN)\cite{7} or ResNet\cite{8} architecture is used to extract speaker embeddings. As the depth of the network increases, the network saturates, resulting in a rapid drop in accuracy. Therefore, ResNet is often used to prevent network degradation. The Probabilistic Linear Discriminant Analysis (PLDA)\cite{9} or the cosine similarity is usually used as the backend for speaker retrieval. Recently, many new architectures are proposed focusing on improving the speaker embedding to achieve a higher retrieval rate, such as the Emphasized Channel Attention, Propagation and Aggregation in Time Delay Neural Network (ECAPA-TDNN)\cite{10}. It adopts the Res2Net\cite{11} and several Squeeze and Excitation (SE)\cite{12} blocks to achieve the state-of-the-art performances in speaker retrieval.

In this paper, we propose several enhancements to the network based on ECAPA-TDNN architecture. The contributions of our work is represented as follows. First, the TDNN network is modified by a concatenation of multi-scale convolution layers processed by multi-scale convolution kernels to achieve better retrieval accuracy. Second, additional residual blocks between SE-Res2blocks are used to prevent deep network from overfitting complex parameters. Third, the attentive statistic pooling is replaced by a combination of several different poolings to improve the results even further.

The paper is organized as follows: Section 2 describes the well-known model architectures for extracting speaker embeddings and some applications in speaker retrieval. Section 3 describes the model architecture proposed by us. Section 4 states the datasets we used, data augmentation recipes and experimental setups. Section 5 describes the results of our experiments and the necessity of each part in the structure of our proposed model. Section 6 depicts the conclusions drawn from the experiments.

2. Related Work

Speaker embeddings and back-end scoring are two significant procedures in speaker retrieval tasks. The frame-level mel-spectrogram features of fixed-length utterances are passed through a deep learning network to provide speaker embeddings for backend. PLDA and cosine similarity are two back-end scoring methods usually adopted in speaker retrieval.

2.1. Speaker Embeddings

One model widely used for extracting speaker embeddings is based on x-vector\cite{6}. In x-vector architectures, the frame-level filterbank features are fed into a 5-layer TDNN network. Then,
the statistic pooling aggregates the frame-level output and computes its mean and standard deviation. After aggregating, the parameters of the network are propagated through two segment-level layers and finally the softmax output layer. The x-vector embeddings are extracted from the first segment-level layer. It presents a strong representation of embeddings in the filed of speaker retrieval. In addition, ResNet [13] is also employed to replace the TDNN network. It has a far-reaching application in extracting speaker embeddings. The internal residual blocks in ResNet use skip connections to alleviate the gradient disappearance problem caused by increasing depth in deep neural networks.

Another state-of-the-art model is the ECAPA-TDNN model [10] which has far-ranging applications in speaker retrieval and identification. First, the TDNN initial frame layers are replaced by 1-D Res2Net module and the SE blocks applied in the image classification are also introduced in these initial frame layers. The SE blocks expand the temporal context of the frame layer by rescaling the channel according to the recorded global properties. Second, the network aggregates the information of each SE block to learn a better representation. Finally, the statistical pooling module is replaced using channel-dependent frame attention, enabling the network to focus on different subsets of frames during statistical estimation for each channel. Batch normalization and fully-connected layers are implemented at last.

These models which are composed of convolution neural networks (CNN) have been used as the front-end of extracting features for speaker embedding. Different poolings can obtain different speaker embeddings. Several pooling methods are proposed to resolve tasks in image retrieval [14], including the sum pooling of convolution (SPoC) [15], global max pooling, statistic pooling [16], attentive pooling [17], global average pooling [18], etc. These methods with better representations in image retrieval can also be applicable to SR problems. Comparing to the fully connected layers, poolings can better adapt to the convolution structure by forming a compulsory mapping between feature maps and categories, and sum out more spatial information. Furthermore, combining various pooling can capture more information of the feature maps spatially and temporally.

2.2. Speaker Retrieval

Unlike speaker verification tasks, speaker retrieval tasks have a large utterances data pool which includes both target and non-target utterances. The speaker retrieval system is designed to find top-$K$ ($K$ is a constant parameter) candidates for each target speaker according to the similarity scores. Therefore, the retrieval system needs more accurate representations of the speaker embeddings and scores to select the best $K$ candidates of each target utterances than verification system. The scoring metrics are used to measure the space distance between two speaker embedding vectors and present the similarity of two vectors. The higher score means higher probabilities between two utterances from one speaker. The scoring function adopts the PLDA or cosines similarity traditionally. The brief blocks of speaker retrieval system are shown in Figure 1.

3. Proposed Models

3.1. Overview

The whole architecture of the retrieval system is depicted in Figure 2. First, the frame-level log Mel-filterbank features are fed into multi-scale convolution layers to produce multi-scale convolution features. The multi-scale filter embeddings are then concatenated on the dimension of channel features. Second, the concatenated embeddings go through into several SE-Res2Blocks with skip connections additionally added, and then the statistic-attentive pooling and SPoC pooling are combined. Finally, the features are passed into batch normalization and fully-connected softmax layers. The output probabilities are considered as speaker embeddings. The details are described in the following subsections.

3.2. Multi-scale convolution layers

The standard ECAPA-TDNN utilizes a 1-D CNN with the kernel size of 5 as the TDNN layer. However, single-scale convolution filtering limits the extraction of feature embedding and provide limited spatial features. By going through multi-scale convolution layers which are adopted by multi-scale convolution kernels, both the short term and high level features are preserved. This multi-scale filtering method is usually done on raw waveform [19] which provides different scales of features at the same time.

In this work, we replace the simple 1-D CNN with three branches of multi-scale convolutions, where the kernel sizes are determined by convolution kernels. Each branch is composed of several layers of 1-D CNN as is shown in Figure 3. It should be noted that the number of layers for each branch is set to 2. Multi-scale convolution kernels with the size of $[5, 3, 1]$ are provided to extract different scales of embedding features. The stride and padding parameters of different kernels are all set to 1. After two layers of convolution, the features extracted from different branches are obviously different. Multiple feature maps are sliced to the minimum feature size and then all feature maps are concatenated along the dimension of channel. In this way, the proposed multi-scale convolution layers are able to provide multi-scale spatial information for a determined frame-level input.
3.3. Residual blocks in SE-Res2blocks

In the architecture of ECAPA-TDNN, the SE-Res2blocks consist of the SE blocks and Res2Net. These blocks are rather deep networks and composed of complex parameters for gradient propagation and backward computation. Thus, when training such deep networks, network non-convergence occurs frequently and affects feature extraction for speaker embeddings.

In response to this problem, we use the skip connections between SE-Res2blocks to prevent network degradation caused by complex parameters and deep layers in training. The building block is shown in Figure 4 and defined as:

$$ Y = F_{SE-Res2block}(X, W_i) + X $$  

(1)

where $X$ and $Y$ represent the input and the output vectors of few stacked layers. $F_{SE-Res2block}(X, W_i)$ denotes the learning layers of the SE-Res2blocks. $F_{SE-Res2block}(X, W_i) + X$ of the building block represents the skip connection and the element-wise addition. Skip connection introduces no additional parameters as it has the same computation cost as the network before. It should be noted that the dimension of $F_{SE-Res2block}(X, W_i)$ and $X$ must be the same.

By going through several SE-Res2layers, the input and output vectors have the same dimension. The residual network can efficiently solve the model degradation problem of deep neural network.

3.4. Combination of multiple poolings

The basic ECAPA-TDNN model adopts channel-wise and context-wise dependent statistics pooling based on attention mechanism. This pooling method outperforms the traditional statistic pooling as it applies various temporal attention to each channel [10]. However, combination of multi-scale pooling methods can achieve better performances than single-scale pooling [20], which is widely applied in the filed of image retrieval.

In our work, two branches of different kinds of poolings are used after the last convolution layer of the network. One pooling
Figure 5: The combination of attentive statistic pooling and SPoC pooling, where the last convolution layer denotes the 1-D CNN after concatenation of multiple SE-Res2blocks.

method is the statistical attentive pooling method which focuses on channel-wise regions of utterance representation. Another pooling method is the SPoC [15] which activates larger regions on utterance representation and presents global descriptions of convolution features.

Given a fixed-length utterance $U$, the output of the hidden layer $h$ is a $D$ tensor with a dimension of $C \times T$, where $C$ denotes the number of feature maps and $T$ denotes the length of frames. Let $h_t (C \times 1)$ and $h_c (1 \times T)$ denote the hidden tensor of frame $t$ and channel $c$. Thus, the hidden layer $h$ can be a set of $h_c$ defined as:

$$h = [h_1, h_2, ..., h_C]^T, c \in \{1, 2, ..., C\}$$

(2)

Then, the sum pooling method of channel $c$ can be defined as:

$$\psi_{\text{sumpool}}_c(U) = \frac{1}{|h_c|} \sum_{t=1}^{T} h_t$$

(3)

The SPoC pooling vector is then produced by a fully-connected layer and an 1-D batch normalization layer represented as follows:

$$P_{\text{SPoC}} = f_{\text{id-BN}}(W \cdot \psi_{\text{sumpool}}(U))$$

(4)

where $P$ is the pooling function of SPoC, $f_{\text{id-BN}}$ represents the batch normalization function and $W$ represents the weight of fully-connected layer.

Finally, the pooling produced by SPoC and statistic attentive pooling are concatenated along the dimension of channel. These concatenated poolings are then passed into a batch normalization layer to effectively avoid gradient vanishing and improve convergence speed. The structure of the combination of pooling layers is presented in Figure 5.

4. Experiments

4.1. Datasets

The datasets to train our proposed system are the CN-Celeb [21, 22] and Aishell [23]. CN-Celeb dataset includes two subsets: CN-Celeb1 and CN-Celeb2, except the CN-Celeb1 evaluation set. It contains speech from Chinese celebrities and covers 11 genres in real condition, including play, movie, interview, etc. The dataset focuses on large-scale and complex scenarios which is challenging for speaker retrieval. Aishell is an open-source Chinese Mandarin speech corpus which is recorded in a pure environment using high fidelity microphone. The total number of speakers is 400. The SR.dev and SR.test set are provided to evaluate our system by the competition organizer. The details of CN-Celeb and Aishell are mentioned in Table 1.

Table 1: Data profile of the datasets we adopt in SR tasks. It is noted that the CN-Celeb.E in CN-Celeb1 is not used by us in experiments.

<table>
<thead>
<tr>
<th>Stages</th>
<th>Datasets</th>
<th>Speakers</th>
<th>Utterances</th>
<th>Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Train</td>
<td>CN-Celeb1</td>
<td>997</td>
<td>126332</td>
<td>271</td>
</tr>
<tr>
<td></td>
<td>CN-Celeb2</td>
<td>1996</td>
<td>524787</td>
<td>1084</td>
</tr>
<tr>
<td></td>
<td>Aishell</td>
<td>400</td>
<td>141600</td>
<td>165</td>
</tr>
<tr>
<td>Dev</td>
<td>SR.dev (target)</td>
<td>5</td>
<td>5</td>
<td>Not given</td>
</tr>
<tr>
<td></td>
<td>SR.dev (pool)</td>
<td>Not given</td>
<td>20050</td>
<td>Not given</td>
</tr>
<tr>
<td>Test</td>
<td>SR.test (target)</td>
<td>25</td>
<td>25</td>
<td>Not given</td>
</tr>
<tr>
<td></td>
<td>SR.test (pool)</td>
<td>Not given</td>
<td>500250</td>
<td>Not given</td>
</tr>
</tbody>
</table>

4.2. Acoustic Features

In the experiments, we randomly crop the audio files into 3 seconds of fixed-length chunks in training process. The input features are 80-dimensional mel-filterbanks with 25ms length Hamming windows and 10ms window shift. No voice activity detection (VAD) is applied during training. The data augmentation recipes include: Specaugment [24], speed perturbation with rates of 0.95 and 1.05, adding noise and reverberation in Room Impulse Response and Noise Database (RIRs) [25] with the signal-noise ratio (SNR) ranges from 0 to 15. These data augmentation recipes make the training more robust.

4.3. Training settings

The implementation of our experiments is adapted on the Speechbrain [26] platform with PyTorch [27] framework. The speaker embeddings extracted by the acoustic features through the network are 192-D vectors. For the first 20 epochs, the Adam optimizer is used with a learning rate of $1e^{-5}$, the weight decay for scheduling the cycle learning rate is set to $2e^{-6}$, and the training process is performed on the CN-Celeb1 dataset, except the CN-Celeb1 evaluation set. The margin and the scale parameters of the Additive Angular Margin (AAM) Loss [28] are set to 0.2 and 30. Furthermore, a fine-tuning method of adjusting the learning rate to $1e^{-6}$ and training with both CN-Celeb1, CN-Celeb2 and Aishell datasets is applied in the following 20 epochs.

The results for SR task are measured by the Mean Average Precision (mAP) [29] evaluation metric which represents the accuracy of retrieving the right enrolled speaker from the large-scale speaker data pool. The higher the scores of mAP are, the better the system performs.

5. Results

5.1. Comparison with the baseline system

As seen in Table 2, our proposed system gets a better performance than the baseline system in terms of the mAP evaluation.
The performance of our system gets 19.6% and 22.7% on dev set and test set respectively. Our proposed system as a whole is more effective than the state-of-the-art ECAPA-TDNN system.

Table 2: Comparision of mAP for several different systems on dev and test set. MC denotes multi-scale convolution layers, MP denotes combination of multiple poolings and residual denotes the skip connection layers.

<table>
<thead>
<tr>
<th>System</th>
<th>Dev</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline (ECAPA-TDNN)</td>
<td>0.726</td>
<td>0.5773</td>
</tr>
<tr>
<td>Proposed</td>
<td>0.868</td>
<td><strong>0.7085</strong></td>
</tr>
<tr>
<td>Proposed – residual</td>
<td>0.860</td>
<td>0.6844</td>
</tr>
<tr>
<td>Proposed – MP</td>
<td>0.840</td>
<td>0.6439</td>
</tr>
<tr>
<td>Proposed – MC</td>
<td>0.851</td>
<td>0.6755</td>
</tr>
<tr>
<td>ECAPA-2-layer-TDNN</td>
<td>0.765</td>
<td>0.6199</td>
</tr>
</tbody>
</table>

5.2. Ablation study

We also adopt ablation study to prove each part of our proposed system is effective. The whole architecture is an assemble of the multi-scale convolution layers, residual learning blocks and the combination of multiple pooling layers. We eliminate these three components one by one to see the performance of our model.

As seen in Table 2, first, the residual blocks are removed. The mAP can achieve to 0.860 in dev set and 0.6844 in test set, where the score descends a little. The residual blocks can slightly improve the network and have a good effect on preventing overfitting. Second, the combination of SPoC pooling and statistic attentive pooling is removed and the system is trained with single statistic attentive pooling. It is shown that our proposed system relatively outperforms the system with no SPoC pooling by 15.7% in dev set and 11.5% in test set. Third, the multi-scale convolution layers are replaced by the traditional TDNN layer which is the same as single 1-D convolution. The multi-scale convolution layers lead to a relative improvement of 17.2% and 17.1% in dev and test set. One may argue that two layers of convolutions can definitely outperform the 1-layer TDNN. Therefore, we also experiment the baseline system with a TDNN of two layers. The performance of ECAPA-2-layer-TDNN is not as good as the proposed multi-scale convolution layers. However, it also achieves a slightly relative improvement of 5.3% and 7.4% in dev and test sets. Overall, the results represent that the variant architectures proposed by us perform great improvements and the most effective method in the whole proposed architecture is the combination of multiple poolings.

5.3. Qualitative analysis

In order to prove the performance of clustering, we adopt t-SNE [30] for visualizing high-dimensional data. The t-SNE can reveal several cluster structures at different scales. It can both reserve the local and global structure. Thus, the clustering of points in the figure is more dense, indicating that the class is more similar.

Figure 6 shows five clusters of utterance-level speaker embeddings which are randomly selected from the training dataset. The scatters of the right figure are more clustered. It proves that our proposed system can better compact intra-speakers and separate inter-speakers. It can lead to an assumption that multi-scale convolution layers, residual blocks in the middle of the network and combination of multiple poolings can make the system more robust and better at clustering.

6. Conclusion

In this paper, we propose an upgraded architecture modified by ECAPA-TDNN to obtain more effective utterance-level speaker embeddings for speaker retrieval. First, to adopt multi-scale feature extraction, multi-scale convolution layers are applied to substitute the single-scale TDNN layer. Second, the residual layers in SE-Res2blocks can better solve the degenerate problems in deep networks. Third, concatenating the SPoC and statistic attentive pooling can provide more spatial information than single-scale pooling. All steps of our enhancements achieve better performances comparing to the ECAPA-TDNN baseline system. Finally, it is found that all proposed components have better inter-class separation and intra-class compactness. The proposed modules can also be used in other speaker embedding extractors, such as ResNet, to improve their network performances. In future, we will verify this assumption.

7. References


