Speaker Diarization with Deep Speaker Embeddings for DIHARD Challenge II

Sergey Novoselov¹,², Aleksei Gusev¹,², Artem Ivanov², Timur Pekhovsky², Andrey Shulipa¹, Anastasia Avdeeva¹,², Artem Gorlanov², Alexandr Kozlov²

¹ITMO University, St.Petersburg, Russia
²STC-innovations Ltd., St.Petersburg, Russia

{novoselov, gusev-a, ivanov-ar, tim, shulipa, avdeeva-a, gorlanov, kozlov-a}@speechpro.com

Abstract

This paper describes the ITMO University (DI-IT team) speaker diarization systems submitted to DIHARD Challenge II. As with DIHARD I, this challenge is focused on diarization task for microphone recordings in varying difficult conditions. According to the results of previous DIHARD I Challenge state-of-the-art diarization systems are based on x-vector embeddings. Such embeddings are clustered using agglomerative hierarchical clustering (AHC) algorithm by means of PLDA scoring. Current research continues the investigation of deep speaker embedding efficiency for the speaker diarization task. This paper explores new types of embedding extractors with different deep neural network architectures and training strategies. We also used AHC to perform embeddings clustering. Alternatively to the PLDA scoring in our AHC procedure we used discriminatively trained cosine similarity metric learning (CSML) model for scoring. Moreover we focused on the optimal AHC threshold tuning according to the specific speech quality. Environment classifier was preliminary trained on development set to predict acoustic conditions for this purpose. We show that such threshold adaptation scheme allows to reduce diarization error rate compared to common AHC threshold for all conditions.

Index Terms: speaker diarization, x-vectors, c-vectors, AHC, PLDA, CSML

1. Introduction

Speaker diarization is the problem of clustering a conversation into segments spoken by the same speaker. Nowadays diarization task for distant/far-field audio under noisy conditions is of particular interest because of its increasing practical significance for automatic voice services.

First DIHARD speech diarization challenge was intended to provide a standard set of data drawn from diverse and challenging conditions to evaluate state-of-the-art diarization system performance and provide a standard set for research.

Similar to the first challenge, DIHARD Challenge II [1, 2, 3] deals with hard recording conditions: far-field microphone or microphone array, low signal-to-noise ratio (SNR) and high percentage of overlapped speech.

This paper describes the ITMO University (DI-IT team) speaker diarization systems submitted to DIHARD Challenge II.

Due to hard recording conditions for direct speaker clustering it is reasonable to use new high-level speaker embeddings extracted from deep neural network (DNN) rather than conventional features used for diarization task not so far ago - like raw mel-frequency cepstral coefficients (MFCC) [4] or, even, i-vectors [5]. Such deep neural network speaker embedding extractors, like x-vectors are successfully used in speaker verification [6] and speaker diarization [7] tasks. They are typically trained on large amount of data, which include augmented data and noisy conditions and can extract speaker embeddings even from highly noised recordings. For this reason we decided not to use clustering methods with a strong prior, such as [8, 9] or in [10]. We applied agglomerative hierarchical clustering (AHC) in deep speaker embeddings space.

According to the results of previous studies on text-independent speaker recognition in telephone [11] and microphone channels [12], deep speaker embeddings based systems (like x-vectors) significantly outperform conventional i-vector based systems in terms of speaker recognition performance. In addition, recent studies [13, 14, 15] present the successful implementation of some proven approaches from face recognition field for deep speaker embeddings extractors training. A comparative study of different back-end solutions for DNN based speaker embeddings was presented in [16]. This work demonstrated that cosine similarity metric learning (CSML) approach can be effectively used for speaker verification in deep neural network (DNN) embeddings domain. It was shown that the performance of deep speaker embeddings based systems can be improved by using CSML with the triplet loss training scheme in both clean and in-the-wild conditions.

During the diarization challenge, we explored several systems based on different deep speaker embeddings extractors [17]. As a back-end scoring models for AHC clustering procedure we used standard Probabilistic discriminant analyses (PLDA) and CSML approach.

We also investigated AHC in the space of the fused PLDA and CSML scores [16]. Our approach for systems fusion was based on Random Forest [18] binary classifiers.

Additionally in this paper we focused on AHC threshold tuning according to the specific speech quality. Environment classifier was preliminary trained on development set to predict acoustic conditions for this purpose.

DIHARD Challenge II is conventionally divided into single channel task and multichannel task. Each task contains two tracks: diarization using reference SAD, diarisation using system SAD. In this paper we present our systems and their results for the single channel task only with reference SAD.

2. Systems description

2.1. Front-End

In the proposed diarization systems we used 40 dimensional MFCC extracted from raw audio signal (16000 Hz) with 25ms
frame-length and 15 ms overlap.

After the features were extracted we applied local CMN over a 3-second sliding window and global Cepstral Mean and Variance Normalization (CMVN) over the whole utterance.

2.2. Speaker representations

In this work we focused on two types of deep neural network speaker embeddings for the diarization task: x-vectors and Speaker Residual Net based embeddings recently proposed by the authors [13]. We refer to the latter as c-vectors.

Our x-vector systems were mainly based on the configuration described in [19] and its modifications. The speaker embeddings in this case are extracted from the affine layer on top of the statistics pooling layer of the classifier network. All x-vector systems for this challenge utilized Kaldi Toolkit [20].

C-vector system for this challenge utilized Pytorch [21]. Our c-vector system was mainly based on the configuration described in [13, 16].

2.3. Probabilistic linear discriminant analysis

The PLDA is successfully used in speaker recognition to specify a generative model of the embeddings presentation. It is assumed that a speaker embedding can be modeled as:

\[ x = m + Vy + \epsilon \]  

where \( m \) is the mean of embeddings, \( y \) denotes the speaker-dependent latent variable with standard normal prior, and \( \epsilon \) is the normally distributed residual noise with zero mean and precision \( \Lambda \). Expectation-maximization (EM) algorithm is used to estimate the parameters of the PLDA model \((V, \Lambda)\) as presented in [22]. After the PLDA model is trained on the development set it can be used in speaker recognition.

The PLDA model makes it possible to calculate the marginal likelihood for target and impostor hypothesis, and correspondingly the PLDA score:

\[ S(x_1, x_2) = \ln \frac{P(x_1, x_2|\text{tar})}{P(x_1|m)p} \cdot P(x_2|m)p \]  

(2)

Worth mentioning that the PLDA backend model is successfully used for speaker diarization to perform AHC clustering procedure [23].

2.4. Cosine similarity metric learning

The discriminative metric learning approach can be viewed as a generative model of the embeddings presentation. It is assumed that a speaker embedding can be modeled as:

\[ x = m + Vy + \epsilon \]  

where \( m \) is the mean of embeddings, \( y \) denotes the speaker-dependent latent variable with standard normal prior, and \( \epsilon \) is the normally distributed residual noise with zero mean and precision \( \Lambda \). Expectation-maximization (EM) algorithm is used to estimate the parameters of the PLDA model \((V, \Lambda)\) as presented in [22]. After the PLDA model is trained on the development set it can be used in speaker recognition.

The PLDA model makes it possible to calculate the marginal likelihood for target and impostor hypothesis, and correspondingly the PLDA score:

\[ S(x_1, x_2) = \ln \frac{P(x_1, x_2|\text{tar})}{P(x_1|m)p} \cdot P(x_2|m)p \]  

(2)

Worth mentioning that the PLDA backend model is successfully used for speaker diarization to perform AHC clustering procedure [23].

**Algorithm 1: Cosine Similarity Metric Learning**

**Input:**
- \((X, Y) = \{x_i, y_i\}_{i=1}^N\) : a set of training samples
- \(d\) : dimension of embeddings

**Output:**
- \(A\) : transformation matrix

```
1: \(A \leftarrow I\) // initialization by the identity matrix
2: while iter \(\leq\) num_iters do
3:   \(\mathcal{L} \leftarrow 0\)
4:   while \(a \leq \text{num} \cdot \text{batches do}\)
5:     \(S_{a,b} \leftarrow CS(X_{b,a}, X, A)\)
6:     \(d_{a,b} \leftarrow f(y_{a,b}, Y, S_{a,b})\)
7:     \(\mathcal{L} \leftarrow \mathcal{L} + \log(1 + \exp(-d_{a,b,k}))\)
8:   end
9:   \(A \leftarrow \arg \min_A \mathcal{L}(A)\)
10: end
```

We optimize (4) with regard to matrix \(A\) by using Adam optimizer implemented in the publicly-available Tensorflow framework [25]. Matrix \(A\) is initialized by the identity matrix. At each optimization step, a triplet loss is formulated by sampling a batch of the training set. The optimization settings are as follows: a batch size of 50, a learning rate of \(10^{-4}\). To ensure an upper triangular view of model \(A\) we apply masking of elements under diagonal. Inputs of the algorithm are pairs of embeddings \(x \in \mathbb{R}^d\) and speaker labels \(y \in \mathbb{N}\) from a training set. For each anchor \(a\) within a batch we calculate similarities \(S_{a,b}\) and split \(f(\cdot)\) them into \(s_{a,b}^+\) positive and \(s_{a,b}^-\) negative subsets according to the speaker labels \(y_{a,b}\). Using the set of relative differences \(d_{a,b}\) between all elements of the subset we obtain objective loss \(\mathcal{L}\) that has to be optimized to train \(A\). The summation in \(\mathcal{L}\) is over all the elements in \(d_{a,b}\). The number of the differences is defined as \(K_{a,b} = N^+_a \cdot N^-_b\) where \(N^+_a, N^-_b\) are the numbers of positive and negative scores in \(s_{a,b}^+\).

2.5. AHC-clustering

The most common approach for speaker diarization task proposed in [23] is using AHC of acoustic segments. Speech segments are clustered together according to similarity metrics (like PLDA or CSML scoring), until the stopping criterion is reached. Usually this is performed by global AHC threshold implementation. It can be tuned on the development set by diarization performance optimization.

2.6. System Fusion

We perform diarization systems fusion on the score level. An ensemble of decision trees [18] was used for this purpose. Each
ensemble is a binary classifier which operates with a vector $s$ of stacked scores from different systems. And in this way it distinguishes target/impostor pairs. By using different weights for target classes it is possible to train set of classifiers and the final score of $N$ different classifiers is calculated as follows:

$$S(s) = \ln \left( \frac{\sum_{n \in N} P_n^{\text{tar}}(s) + \epsilon}{\sum_{n \in N} P_n^{\text{imp}}(s) + \epsilon} \right)$$

where $P_n^{\text{tar}}$ and $P_n^{\text{imp}}$ are the output of the $n$-th binary classifier voted for target and impostor class respectively.

2.7. Recording condition detector (RCD)

Development and evaluation data from single channel track is represented by 5-10 minute duration samples related to 11 different acoustic recording conditions. In this investigation we trained acoustic environment detector based on standard x-vector classifier with softmax activation. Our RCD uses MFCC stack of each segment as input. Classifier was trained on the development set. In addition we performed development data augmentation (babble, noise, music, reverberation) and fragments random sampling during training process. Total training data contains approximately 100 hours of speech.

Classifiers for 11 and 5 pooled acoustic condition classes were explored. We tuned AHC thresholds for each class separately and used these thresholds on the evaluation step after automatic acoustic conditions classification.

3. Implementation details

In this work we focused on two types of deep neural network speaker embeddings for the diarization task: x-vectors and c-vector [17]. X-vector based systems are:

- Xvec-TDNN: Standard x-vector system described in [26].
- Xvec-Ext-TDNN: Configuration of this system was an extended version of the original TDNN system used in Xvec-TDNN. The differences here include an additional TDNN layer with wider temporal context, and unit context TDNN layers between wide context TDNN layers. This approach is taken from the JHU-MIT System Description for NIST SRE18. Moreover, before StatPooling layer we used LSTM-layer with cell dimension of 512, delay in the recurrent connections equal to -3, and both recurrent and non-recurrent projection dimension equal to 256. The LSTM layer context was reduced to 3.
- Xvec-Ext-TDNN-LSTM: Configuration of this system is an extended version of the original TDNN system used in Kaldi [20]. The differences here include an additional TDNN layer with wider temporal context, and unit context TDNN layers between wide context TDNN layers. This approach is taken from the JHU-MIT System Description for NIST SRE18. Moreover, before StatPooling layer we used LSTM-layer with cell dimension of 512, delay in the recurrent connections equal to -3, and both recurrent and non-recurrent projection dimension equal to 256. The LSTM layer context was reduced to 3.

Our c-vector embedding architecture [17] is based on residual blocks built using TDNN architecture, MFM (Max-Feature-Map) activations [28] and A-Softmax (Angular Softmax) activation [29].


4. Training data

According to the last studies [19, 30, 13] training data preparation plays a crucial role in deep speaker embeddings extractor training. Therefore, in this work we paid great attention to the selection of training data for tuning speaker diarization systems in single channel far-field audio, under noisy conditions. The recordings sampling rate was 16000 Hz.

Training corpus includes VoxCeleb1, VoxCeleb2 and SITW and their augmented versions. Augmented data was generated using part of standard augmentation recipe from Kaldi Toolkit [26] using the freely available MUSAN and RIR datasets. Augmentation was performed in order to simulate the distortions typical to far-field microphone under noisy conditions. Reverberation was performed using the impulse response generator based on [31]. Four different RIRs were generated for each of 40,000 rooms with a varying position of sources and destructors. It should be noted that, in contrast to the original Kaldi augmentation, we reverberated both speech and noise signals. In this case different RIRs generated for one room were used for speech and noise signals respectively. Thus we obtained more realistic data augmentation. The final database consists of approximately 5,200,000 examples (7562 speakers). Energy-based SAD from Kaldi Toolkit [26] was applied to select speech frames from the data. Audio samples with speech duration less than 3.5 seconds were excluded and the maximum amount of samples for one speaker was limited to 8.

5. Experimental results

Experiment results for our single and fusion systems on the development and evaluation sets are presented in Table 1 in terms of Diarization Error Rate (DER). This results demonstrate the efficiency of DNN-based speaker embedding extractors for speaker diarization task in single channel distant/far-field audio under noisy conditions.

It should be noted that baseline Kaldi Toolkit diarization system (Xvec-TDNN with PLDA scoring model) corresponds to B1 in Table 1. This system achieves $DER = 23.95\%$ on the development set. Our alternative single C-vector based system (S0) is inferior than X-vector (B1) and leads to $DER = 24.10\%$ on the development set.

It is worth mentioning that deeper x-vector extractors with...
In this paper we investigated the single channel diarization lower than DER of the S3-S4. For example, DER of the S5-S9 fusion systems is almost rate compared to common AHC threshold for all conditions. Threshold adaptation scheme allows to reduce diarization error results in Table 1)

1% vector based systems improves diarization quality by

Table 1: Results of our single and fusion systems on the development and evaluation sets

<table>
<thead>
<tr>
<th>Name</th>
<th>Extractor</th>
<th>Backend</th>
<th>Number of condition clusters</th>
<th>DER [%]</th>
<th>dev</th>
<th>eval</th>
</tr>
</thead>
<tbody>
<tr>
<td>B1</td>
<td>Xvec-TDNN</td>
<td>PLDA</td>
<td>1</td>
<td>22.95</td>
<td>22.12</td>
<td></td>
</tr>
<tr>
<td>S0</td>
<td>Cvec-Wide-Res-TDNN</td>
<td>PLDA</td>
<td>1</td>
<td>24.10</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>S1</td>
<td>Xvec-Ext-TDNN-LSTM</td>
<td>PLDA</td>
<td>1</td>
<td>22.24</td>
<td>24.61</td>
<td></td>
</tr>
<tr>
<td>S2</td>
<td>Xvec-Ext-TDNN-LSTM</td>
<td>PLDA</td>
<td>1</td>
<td>22.08</td>
<td>24.44</td>
<td></td>
</tr>
<tr>
<td>S3</td>
<td>Xvec-TDNN</td>
<td>PLDA</td>
<td>1</td>
<td>21.86</td>
<td>23.74</td>
<td></td>
</tr>
<tr>
<td>S4</td>
<td>Xvec-TDNN</td>
<td>PLDA</td>
<td>1</td>
<td>21.54</td>
<td>22.51</td>
<td></td>
</tr>
<tr>
<td>S5</td>
<td>Xvec-TDNN</td>
<td>PLDA</td>
<td>1</td>
<td>20.12</td>
<td>21.88</td>
<td></td>
</tr>
<tr>
<td>Cvec-Wide-Res-TDNN</td>
<td>CSML</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>S6</td>
<td>Xvec-TDNN</td>
<td>PLDA</td>
<td>5</td>
<td>20.54</td>
<td>21.62</td>
<td></td>
</tr>
<tr>
<td>S7</td>
<td>Xvec-TDNN</td>
<td>PLDA</td>
<td>11</td>
<td>19.74</td>
<td>22.4</td>
<td></td>
</tr>
<tr>
<td>Cvec-Wide-Res-TDNN</td>
<td>CSML</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>S8</td>
<td>Xvec-TDNN</td>
<td>PLDA</td>
<td>5</td>
<td>19.92</td>
<td>22.22</td>
<td></td>
</tr>
<tr>
<td>S9</td>
<td>Xvec-TDNN</td>
<td>PLDA</td>
<td>11</td>
<td>20.24</td>
<td>22.02</td>
<td></td>
</tr>
<tr>
<td>Cvec-Wide-Res-TDNN</td>
<td>CSML</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

additional LSTM frame level layer perform better than original x-vector system in the speaker diarization task. Using extended deep neural network architecture Xvec-Ext-TDNN-LSTM for speaker embedding extractor helps to decrease DER to 22.24% on the development set.

According to our observations diarization performance of the PLDA-based and CSML-based systems scoring are close. But such systems fusion results in diarization quality improvement (S2 system result is \( \text{DER} = 22.08\% \)).

As well we found out that the fusion of X-vector and C-vector based systems improves diarization quality by 1% in terms of DER on the evaluation set (compare S3 and S4 system results in Table 1)

It should be also pointed out that proposed condition AHC threshold adaptation scheme allows to reduce diarization error rate compared to common AHC threshold for all conditions. For example, DER of the S5-S9 fusion systems is almost 2% lower than DER of the S3-S4.

6. Conclusion

In this paper we investigated the single channel diarization problem from Dihard II Challenge. We explored different deep speaker embeddings extractors based on x-vector and c-vector approaches for this purpose. We manage to reduce systems DER by using deeper and extended speaker embeddings extractors combined with different backend scoring models. Moreover we focused on the optimal AHC threshold tuning according to the specific speech quality. Environment classifier was trained on the development set to predict acoustic conditions for this purpose. Such threshold adaptation scheme reduces diarization error rate compared to common AHC threshold for all conditions.

7. Acknowledgements

This work was partially financially supported by the Government of the Russian Federation (Grant 08-08) and by the Foundation NTI (contract 20/18gr) ID 0000000007418QR20002.

8. References


