INTERSPEECH 2019
September 15–19, 2019, Graz, Austria

Speaker Diarization with Deep Speaker Embeddings for DIHARD Challenge II

Sergey Novoselov[1,2], *Aleksei Gusev*[1,2], *Artem Ivanov*[2], *Timur Pekhovsky*[2], *Andrey Shulipa*[1], *Anastasia Avdeeva*[1,2], *Artem Gorlanov*[2], *Alexandr Kozlov*[2]

[1]ITMO University, St.Petersburg, Russia
[2]STC-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 the previous DIHARD I Challenge state-of-the-art diarization systems are based on x-vector embeddings. Such embeddings are clustered using aglomerative 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 analisys (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 mulichannel 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

Copyright © 2019 ISCA

http://dx.doi.org/10.21437/Interspeech.2019-2757

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:

$$\mathbf{x} = \mathbf{m} + \mathbf{V}\mathbf{y} + \boldsymbol{\epsilon} \tag{1}$$

where $\mathbf{m}$ is the mean of embeddings, $\mathbf{y}$ denotes the speaker-dependent latent variable with standard normal prior, and $\boldsymbol{\epsilon}$ is the normally distributed residual noise with zero mean and precision $\boldsymbol{\Lambda}$. Expectation-maximization (EM) algorithm is used to estimate the parameters of the PLDA model $(\mathbf{V}, \boldsymbol{\Lambda})$ 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 imposter hypothesis, and correspondingly the PLDA score:

$$\mathcal{S}(\mathbf{x_1}, \mathbf{x_2}) = \ln \frac{P(\mathbf{x_1}, \mathbf{x_2}|tar)}{P(\mathbf{x_1}|imp) \cdot P(\mathbf{x_2}|imp)} \tag{2}$$

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 an alternative to simple cosine metric or PLDA backend for deep speaker embeddings.

According to the formulation of the CSML, a linear transformation $\mathbf{A}$ must be learned to compute cosine similarities (CS) on a pair $(\mathbf{x_1}, \mathbf{x_2})$ as follows:

$$\mathcal{S}(\mathbf{x_1}, \mathbf{x_2}, \mathbf{A}) = \frac{(\mathbf{A}\mathbf{x_1})^T(\mathbf{A}\mathbf{x_2})}{\|\mathbf{A}\mathbf{x_1}\|\|\mathbf{A}\mathbf{x_2}\|} \tag{3}$$

where the transformation matrix $\mathbf{A}$ is upper triangular. Under this constraint $\mathbf{A}^T\mathbf{A}$ is positive-definite. Unlike [24] we set the triplet loss objective function for training $\mathbf{A}$:

$$\begin{aligned} \mathcal{L}(\mathbf{A}) &= \sum_{a,p,n \in T} \log(1 + \exp(-d_{a,p,n})) \\ \mathbf{A} &= \arg\min_{\mathbf{A}} \mathcal{L}(\mathbf{A}) \end{aligned} \tag{4}$$

where $d_{a,p,n} = s_{a,p} - s_{a,n}$ is the difference between similarity scores $s_{a,p}$ and $s_{a,n}$. T is a collection of training triplets which is formed from a training dataset. A triplet (a, p, n) contains an anchor sample a as well as a positive $p \neq a$ and a negative n example of the anchor's identity. As it can be seen, the minimization of $\mathcal{L}$ increases the relative margin between positive and negative examples, that makes for reducing recognition error on training and evaluation sets.

The metric learning algorithm is presented below:

Algorithm 1: Cosine Similarity Metric Learning

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

Output:
- A : transformation matrix

$A \leftarrow I$ // initialization by the identity matrix
while $iter \leqslant num\_iters$ **do**
  while $b \leqslant num\_batches$ **do**
    $\mathcal{L} = 0$
    while $a \leqslant bsize$ **do**
      $\mathbf{S_{a,b}} \leftarrow CS(\mathbf{x_{b,a}}, \mathbf{X}, \mathbf{A})$
      $\mathbf{s^+_{a,b}}, \mathbf{s^-_{a,b}} \leftarrow f(y_a, \mathbf{Y}, \mathbf{S_{a,b}})$
      $\mathbf{d_{a,b}} \leftarrow \mathbf{s^+_{a,b}} - \mathbf{s^-_{a,b}}$
      $\mathcal{L} += \sum_{k \leqslant K_{a,b}} \log(1 + \exp(-d_{a,b,k})$
    end
    $\mathbf{A} \leftarrow \arg\min_{\mathbf{A}} \mathcal{L}(\mathbf{A})$
  end
end

We optimize (4) with regard to matrix A by using Adam optimizer implemented in the publicly-available Tensorflow framework [25]. Matrix $\mathbf{A}$ 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 matrix $\mathbf{A}$ we apply masking of elements under diagonal. Inputs of the algorithm are pairs of embeddings $\mathbf{x} \in \mathcal{R}^d$ and speaker labels $y \in \mathcal{N}$ from a training set. For each anchor a within a batch we calculate similarities $\mathbf{S_{a,b}}$ and split $f(\cdot)$ them into $\mathbf{s^+_{a,b}}$ positive and $\mathbf{s^-_{a,b}}$ negative subsets according to the speaker labels $y_a, \mathbf{Y}$. Using the set of relative differences $\mathbf{d_{a,b}}$ between all elements of the subsets we obtain objective loss $\mathcal{L}$ that has to be optimized to train $\mathbf{A}$. The summation in $\mathcal{L}$ is over all the elements in $\mathbf{d_{a,b}}$. The number of the differences is defined as $K_{a,b} = N^+_{a,b} \cdot N^-_{a,b}$, where $N^{+/-}_{a,b}$ are the numbers of positive and negative scores in $\mathbf{s^{+/-}_{a,b}}$.

2.5. AHC-clustering

The most common approach for speaker diarisation 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 $\mathbf{s}$ 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:

$$\mathcal{S}(\mathbf{s}) = \ln\left(\frac{\sum_{n \in N} P_n^{tar}(\mathbf{s})}{\sum_{n \in N} P_n^{imp}(\mathbf{s})} + \epsilon\right) \tag{5}$$

where P_n^{tar} and P_n^{imp} are the output of the n-th binary classifier voted for target and imposter 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 acivation. 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 [27].

Xvect-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].

Cvec-Wide-ResTDNN: uses 20 Extended ResTDNN blocks with fixed parameter f = 4.

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.

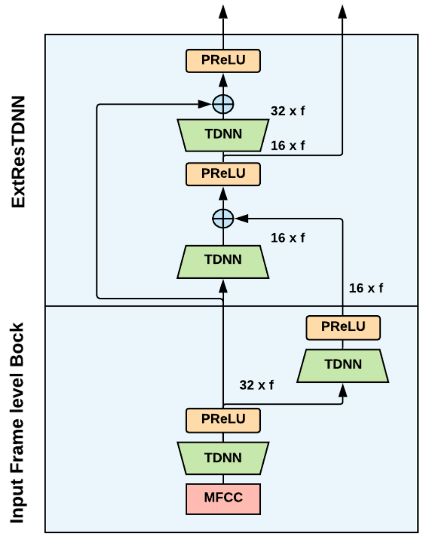

Figure 1: *Residual block structure, used in C-vector based systems. Here **f** denotes the fixed parameter, which defines the size of layers used in Extended ResTDNN blocks.*

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 [1]. 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

[1] http://www.openslr.org

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

Name	Extractor	Backend	Number of condition clusters	DER, [%]	
				dev	eval
B1	Xvec-TDNN	PLDA	1	23.95	25.32
S0	Cvec-Wide-ResTDNN	PLDA	1	24.10	-
S1	Xvec-Ext-TDNN-LSTM	PLDA	1	22.24	24.67
S2	Xvec-Ext-TDNN-LSTM	PLDA	1	22.08	24.44
	Xvec-Ext-TDNN-LSTM	CSML			
S3	Xvec-TDNN	PLDA	1	21.86	23.74
	Xvec-Ext-TDNN	PLDA			
	Xvec-Ext-TDNN-LSTM	PLDA			
	Xvec-Ext-TDNN-LSTM	CSML			
S4	Xvec-TDNN	PLDA	1	21.54	22.51
	Xvec-Ext-TDNN	PLDA			
	Xvec-Ext-TDNN-LSTM	PLDA			
	Xvec-Ext-TDNN-LSTM	CSML			
	Cvec-Wide-ResTDNN	COS			
S5	Xvec-TDNN	PLDA	11	20.12	21.88
	Xvec-Ext-TDNN	PLDA			
	Xvec-Ext-TDNN-LSTM	PLDA			
	Xvec-Ext-TDNN-LSTM	CSML			
	Cvec-Wide-ResTDNN	COS			
S6	Xvec-TDNN	PLDA	5	20.54	21.62
	Xvec-Ext-TDNN	PLDA			
	Xvec-Ext-TDNN-LSTM	PLDA			
	Xvec-Ext-TDNN-LSTM	CSML			
	Cvec-Wide-ResTDNN	COS			
S7	Xvec-TDNN	PLDA	11	19.74	22.4
	Xvec-Ext-TDNN	PLDA			
	Xvec-Ext-TDNN-LSTM	PLDA			
	Xvec-TDNN	CSML			
	Xvec-Ext-TDNN	CSML			
	Xvec-Ext-TDNN-LSTM	CSML			
	Cvec-Wide-ResTDNN	COS			
S8	Xvec-TDNN	PLDA	5	19.92	22.22
	Xvec-Ext-TDNN	PLDA			
	Xvec-Ext-TDNN-LSTM	PLDA			
	Xvec-TDNN	CSML			
	Xvec-Ext-TDNN	CSML			
	Xvec-Ext-TDNN-LSTM	CSML			
	Cvec-Wide-ResTDNN	COS			
S9	Xvec-TDNN	PLDA	11	20.24	22.02
	Xvec-Ext-TDNN	PLDA			
	Xvec-Ext-TDNN-LSTM	PLDA			
	Xvec-Ext-TDNN-LSTM	COS			
	Cvec-Wide-ResTDNN	COS			

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 $DER = 22.08\%$).

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

[1] N. Ryant, K. Church, C. Cieria, A. Cristia, J. Dud, S. Ganapathy, and M. Liberman, "The Second DIHARD Diarization Challenge: Dataset, task, and baselines," in *Proceedings of INTERSPEECH 2019. ISCA. Graz, Autria*, 2019.

[2] E. Bergelson, "Bergelson Seedlings HomeBank Corpus," in *doi: 10.21415/T5PK6D*, 2016.

[3] N. Ryant, K. Church, C. Cieria, A. Cristia, J. Dud, S. Ganapathy, and M. Liberman, "DIHARD Corpus," in *Linguistic Data Consortium*, 2019.

[4] S. E. Tranter and D. A. Reynolds, "An overview of automatic speaker diarization systems," *IEEE Transactions on audio, speech, and language processing*, vol. 14, no. 5, pp. 1557–1565, 2006.

[5] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788–798, 2011.

[6] D. Snyder, P. Ghahremani, D. Povey, D. Garcia-Romero, Y. Carmiel, and S. Khudanpur, "Deep neural network-based speaker embeddings for end-to-end speaker verification," in *2016 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2016, pp. 165–170.

[7] D. Garcia-Romero, D. Snyder, G. Sell, D. Povey, and A. McCree, "Speaker diarization using deep neural network embeddings," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 4930–4934.

[8] P. Kenny, D. Reynolds, and F. Castaldo, "Diarization of telephone conversations using factor analysis," *IEEE Journal of Selected Topics in Signal Processing*, vol. 4, no. 6, pp. 1059–1070, 2010.

[9] R. Zheng, C. Zhang, S. Zhang, and B. Xu, "Variational bayes based i-vector for speaker diarization of telephone conversations," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2014, pp. 91–95.

[10] S. H. Shum, N. Dehak, R. Dehak, and J. R. Glass, "Unsupervised methods for speaker diarization: An integrated and iterative approach," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 10, pp. 2015–2028, 2013.

[11] S. O. Sadjadi, T. Kheyrkhah, A. Tong, C. S. Greenberg, D. A. Reynolds, E. Singer, L. P. Mason, and J. Hernandez-Cordero, "The 2016 nist speaker recognition evaluation." in *Interspeech*, 2017, pp. 1353–1357.

[12] M. McLaren, L. Ferrer, D. Castan, and A. Lawson, "The 2016 speakers in the wild speaker recognition evaluation," in *INTERSPEECH*, 2016, pp. 823–827.

[13] S. Novoselov, A. Shulipa, I. Kremnev, A. Kozlov, and V. Shchemelinin, "On deep speaker embeddings for text-independent speaker recognition," in *Odyssey 2018 The Speaker and Language Recognition Workshop, June 26-29, Les Sables d'Olonne, France, Proceedings*, 2018, pp. 378–385.

[14] Z. Huang, S. Wang, and K. Yu, "Angular softmax for short-duration text-independent speaker verification," *Proc. Interspeech, Hyderabad*, 2018.

[15] W. Cai, J. Chen, and M. Li, "Exploring the encoding layer and loss function in end-to-end speaker and language recognition system," *arXiv preprint arXiv:1804.05160*, 2018.

[16] S. Novoselov, "Triplet loss based cosine similarity metric learning for text-independent speaker recognition," in *INTERSPEECH 2018 – 19th Annual Conference of the International Speech Communication Association, August 2-6, Hyderabad, India, Proceedings*, 2018, pp. 2242–2246.

[17] Sergey Novoselov, Aleksei Gusev, Artem Ivanov, Timur Pekhovsky, Andrey Shulipa, Galina Lavrentyeva, Vladimir Volokhov, Alexandr Kozlov, "STC Speaker Recognition Systems for the VOiCES From a Distance Challenge," in *INTERSPEECH*, 2019.

[18] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[19] D. Snyder, D. Garcia-Romero, D. Povey, and S. Khudanpur, "Deep neural network embeddings for text-independent speaker verification," in *INTERSPEECH 2017 – 18th Annual Conference of the International Speech Communication Association, August 20-24, Stockholm, Sweden, Proceedings*, 2017, pp. 999–1003.

[20] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz *et al.*, "The kaldi speech recognition toolkit," in *IEEE 2011 workshop on automatic speech recognition and understanding*, no. EPFL-CONF-192584. IEEE Signal Processing Society, 2011.

[21] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," in *NIPS-W*, 2017.

[22] S. J. Prince and J. H. Elder, "Probabilistic linear discriminant analysis for inferences about identity," in *2007 IEEE 11th International Conference on Computer Vision*. IEEE, 2007, pp. 1–8.

[23] G. Sell, D. Snyder, A. McCree, D. Garcia-Romero, J. Villalba, M. Maciejewski, V. Manohar, N. Dehak, D. Povey, S. Watanabe, and S. Khudanpur, "Diarization is hard: Some experiences and lessons learned for the jhu team in the inaugural dihard challenge," 09 2018, pp. 2808–2812.

[24] H. Nguyen and L. Bai, "Cosine similarity metric learning for face verification," in *ACCV 2010 – 10th Asian Conference on Computer Vision, November 8-12, Queenstown, New Zealand, Proceedings*, 2010, pp. 709–720.

[25] Google, "TensorFlow". [Online]. Available: https://www.tensorflow.org

[26] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust DNN embeddings for speaker recognition," in *ICASSP 2018 – IEEE International Conference on Acoustics, Speech and Signal Processing, April 15-20, Calgary, Canada, Proceedings*, 2018, pp. 5329–5333.

[27] NIST, "NIST 2018 speaker recognition evaluation plan," https://www.nist.gov/file/453891, 2018, [Online; accessed 03-October-2018].

[28] X. Wu, R. He, Z. Sun, and T. Tan, "A light CNN for deep face representation with noisy labels," *IEEE Transactions on Information Forensics and Security*, vol. 13, pp. 2884–2896, 2018.

[29] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song, "Sphereface: Deep hypersphere embedding for face recognition," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2017)*, vol. 1. IEEE, 2017, pp. 6738–6746.

[30] M. McLaren, D. Castan, M. K. Nandwana, L. Ferrer, and E. Yilmaz, "How to train your speaker embeddings extractor," 2018.

[31] J. B. Allen and D. A. Berkley, "Image method for efficiently simulating small-room acoustics," *The Journal of the Acoustical Society of America*, vol. 65, no. 4, pp. 943–950, 1979.