Interspeech 2018
2-6 September 2018, Hyderabad

Triplet Loss Based Cosine Similarity Metric Learning for Text-independent Speaker Recognition

Sergey Novoselov[1,2]*,Vadim Shchemelinin*[1,2]*, Andrey Shulipa*[2]*, Alexandr Kozlov*[1] *and Ivan Kremnev*[1]

[1]STC Ltd., St. Petersburg, Russia
[2]ITMO University, St. Petersburg, Russia

{novoselov,shchemelinin,shulipa,kozlov,kremnev}@speechpro.com

Abstract

Deep neural network based speaker embeddings become increasingly popular in the text-independent speaker recognition task. In contrast to a generatively trained i-vector extractor, a DNN speaker embedding extractor is usually trained discriminatively in the closed set classification scenario using softmax. The problem we addressed in the paper is choosing a dnn based speaker embedding backend solution for the speaker verification scoring. There are several options to perform speaker verification in the dnn embedding space. One of them is using a simple heuristic speaker similarity metric for scoring (e.g. cosine metric). Similarly with i-vector based systems, the standard Linear Discriminant Analisys (LDA) followed by the Probabilistic Linear Discriminant Analisys (PLDA) can be used for segregating speaker information. As an alternative, the discriminative metric learning approach can be considered. This work demonstrates that performance of deep speaker embeddings based systems can be improved by using Cosine Similarity Metric Learning (CSML) with the triplet loss training scheme. Results obtained on Speakers in the Wild and NIST SRE 2016 evaluation sets demonstrate superiority and robustness of CSML based systems.

Index Terms: speaker recognition, cosine similarity metric learning, speaker embeddings

1. Introduction

I-vector-based systems are widely recognized as state-of-the-art solutions to the text-independent speaker verification problem [1, 2, 3]. Nonetheless, this problem is gradually gaining attention from the deep learning perspective. Particularly, studies [2, 4] make use of the ASR deep neural network (ASR DNN) in order to divide acoustic space into senone classes, and the classic total variability (TV) model is applied to discriminate between speakers in that space afterwards [1].

In such phonetic discriminative DNN-based systems two major techniques can be distinguished. The first one uses DNN posteriors to calculate Baum-Welch statistics, and the second one uses bottleneck features together with speaker specific features (MFCC) for a full TV-UBM system training.

Deep learning frameworks are a powerful tool for complex data analysis [5, 6, 7, 8, 9], and many researches consider training deep non-linear extractors as a solution to the direct speaker discrimination task. Several solid studies demonstrate the advantages of deep end-to-end solutions for discriminating speakers directly in the text-dependent task [10, 11]. Papers [12, 13] describe a deep network extracting a small speaker footprint that is used to discriminate between speakers.

Paper [14] presents a well-performing implementation of a DNN extractor based on the speaker discriminative approach in the text-independent task. One of the key features of the proposed system is the time-delay neural network architecture of the extractor [15] with a statistics pooling layer designed to accumulate speaker information from the whole speech segment into a single vector called an x-vector. Extracted from an intermediate layer of the neural network which comes after the statiscics pooling layer, x-vectors demonstrate properties similar to those of i-vectors from total variability space, which makes it possible to effectively use them in the standard Linear Discriminant Analysis (LDA) followed by Probabilistic Linear Discriminant Analysis (PLDA) [16, 17, 18] backend for segregating speaker information.

Our recent paper [19] demonstrates two alternative deep speaker extractor configurations which are trained with the help of the margin based angular softmax layer instead of the usual softmax classification layer. The work [19] also presents a well-performing similarity metric learning approach as an alternative to standard LDA-PLDA backend model.

The learning of the distance/similarity metric between pairs of the compared samples and investigation of loss functions have a great importance for a variety of tasks, especially in the visual recognition domain. Most investigations consider metric learning on the linear models [20, 21, 22] because they are more convenient to be optimized and allow to avoid overfitting. The nonlinear metric models are also of the interest [23, 24, 25] and improve the recognition performance on some tasks, but they can be prone to overfitting.

This paper presents an advanced comparative study of applying cosine similarity metric learning (CSML) [26] approach for DNN-based speaker embeddings to discriminate speakers. Unlike [26] we apply the triplet loss objective function in order to train the transformation matrix parameters of cosine similarity metric. We compare the proposed backend technique with commonly used cosine and LDA-PLDA approaches for different types of deep extractors and in different conditions. We evaluate the considered speaker recognition systems on the NIST SRE 2010 det 5 protocol, the NIST SRE 2016 and the Speaker-in-the-Wild challenge protocols.

2. Deep speaker embeddings

Recent works [27, 14, 19] show successful implementations of the discriminatively trained deep non-linear extractors in the text-independent speaker discrimination task. According to our observations the two main differences between DNN-based speaker embedding extractors are neural network architectures and their training strategies. This section briefly describes three types of extractors we investigate in the paper. All of them take 23 mel-frequency cepstral coefficients as input [19] and are based on utilizing statistics pooling layer [14] to accumulate speaker specific information over time.

10.21437/Interspeech.2018-1209

2.1. X-vectors

The X-vector system is based on a successful implementation of a deep neural network extractor (X-vectorNet) of the speaker specific information presented in [14]. The regular softmax cross entropy loss function is used to train X-vector extractor by using natural-gradient (NG) modification [28] of the stochastic gradient descend (SGD) algorithm.

Frame layers constitute the time-delay deep neural network (TDNN) part of the system that extracts high-level features from the input signal with a gradually expanding context. After that, the stats pooling layer folds features along the time axis by aggregating their mean and standard deviation statistics. Eventually, these statistics are forwarded to segment levels of the network that employ fully-connected layers to extract an x-vector.

2.2. Max pooling embeddings

The paper [19] demonstrates how an alternative training objective can be used to tune the extractor. More specifically, angular softmax layer with a proper angular margin instead of the usual softmax classification layer can be used to discriminate speakers in the training set. Two different neural network architectures are proposed for speaker embedding extraction. One of them (SpeakerMaxPoolNet) is based on max-pooling layers. This modification of the original TDNN-based extractor [14] includes layer-wise context shrinking when passing from bottom to top layers with addition of max pooling operations at each frame layer. Notably, this approach reduces network size and speeds up computations. One other tweak is the parametrization of the rectified linear unit (ReLU) activation function (PReLU) [29].

The segment-level part of the network is also modified. Here we use Max-Feature-Map (MFM) activation [30] in place of ReLU. In contrast to the commonly used ReLU function that suppresses a neuron by a threshold (or bias), MFM suppresses a neuron by a competitive relationship. By doing so the MFM activation acts as an embedded feature selector.

After the classifier is trained, the last fully-connected layer with its angular softmax activation is removed from the network in order to obtain an extractor of high-level representations for speaker specific information.

We refer the reader to [19] for more details about angular margin softmax and max pooling embeddings system.

2.3. Deep residual embeddings

There are two ways to expand the context in the TDNN architecture: either by widening it at each frame-level layer, or by deepening the network to accumulate richer context with a higher level of feature abstraction. Our deepest architecture for speaker embedding extraction is represented by a deep neural network with TDNN residual blocks (SpeakerResNet) which was also proposed in our previous work [19]. Similarly to max pooling embedding approach the extractor is trained with angular softmax layer.

Our SpeakerResNet architecture is a deep extractor consisting of time-delay layers with shallow frame-level contexts, which are set to 3. The segment-level part of the network is the same as in SpeakerMaxPoolNet. Speaker embeddings are extracted from the last MFM layer.

3. Baseline backend models

In our experiments we investigate the performance of the speaker embeddings with a backend and without one. In the last case simple cosine similarity metric is applied for verification.

Similarly to the i-vectors, LDA followed by PLDA [18] can be used in the DNN-based speaker embedding space.

3.1. Cosine scoring

Simple cosine scoring (1) is very common in biometric verification tasks. When using cosine scoring on centered and usually whitened embeddings $\mathbf{x_1}$, $\mathbf{x_2}$ one measures speaker similarity by computing the correlation coefficient:

$$\mathcal{S}(\mathbf{x_1}, \mathbf{x_2}) = \frac{\mathbf{x_1}^T \mathbf{x_2}}{\|\mathbf{x_1}\| \|\mathbf{x_2}\|} \tag{1}$$

3.2. Probabilistic linear discriminant analysis

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

$$\mathbf{x} = \mathbf{m} + \mathbf{V}\mathbf{y} + \boldsymbol{\epsilon} \tag{2}$$

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 [17]. 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:

$$\mathcal{S}(\mathbf{x_1}, \mathbf{x_2}) = \ln \frac{P(\mathbf{x_1}, \mathbf{x_2} | tar)}{P(\mathbf{x_1} | imp) \cdot P(\mathbf{x_2} | imp)} \tag{3}$$

It should be noted that for speaker recognition tasks the PLDA model performs better when LDA projection and length normalization are used as preprocessing steps [31].

4. Cosine similarity metric learning

The discriminative metric learning approach can be viewed as an alternative to simple cosine metric or LDA-PLDA backend for deep speaker embeddings. According to the formulation of the CSML [26], a linear transformation $\mathbf{A}$ must be learned to compute cosine similarities (CS) on a pair $(\mathbf{x_1}, \mathbf{x_2})$ as follows:

$$\mathcal{S}(\mathbf{x_1}, \mathbf{x_2}, \mathbf{A}) = \frac{(\mathbf{A}\mathbf{x_1})^T (\mathbf{A}\mathbf{x_2})}{\|\mathbf{A}\mathbf{x_1}\| \|\mathbf{A}\mathbf{x_2}\|} \tag{4}$$

where the transformation matrix $\mathbf{A}$ is upper triangular. Under this constraint $\mathbf{A}^T\mathbf{A}$ is positive-definite. Unlike [26] we set the triplet loss objective function for training $\mathbf{A}$:

$$\begin{aligned} \mathcal{L}(\mathbf{A}) &= \sum_{a,p,n \in T} \log(1 + \exp(-d_{a,p,n})) \\ \mathbf{A} &= \arg\min_{\mathbf{A}} \mathcal{L}(\mathbf{A}) \end{aligned} \tag{5}$$

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 (5) with regard to matrix A by using Adam optimizer implemented in the publicly-available Tensorflow framework [32]. 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 $\mathbf{s^{+/-}_{a,b}}$.

Cross validation test is used for early stopping during the training process. As demonstrated in [33] convergence rate of the optimization procedure depends on the ability to choose useful triplets that give a large loss value. To satisfy this condition, we include in the collection of training triplets all positive examples and only 1500 hardest among all negative examples for selected anchors.

5. Experimental setup

5.1. Systems configurations

For the x-vector extractor we follow the configuration presented in [27]. For SpeakerMaxPoolNet and SpeakerResNet systems we follow our previous setups [19].

It should be noted that all speaker embeddings we used have the same dimension of 512. In LDA-PLDA scenario we use LDA to reduce the dimension to 200 and apply a simplified gaussian PLDA model with 200 eigenvoices on the centered and length-normalized embeddings. Our CSML settings are described in Section 4. For the x-vector based system we apply whitening and length normalization of embeddings as the preprocessing steps for CSML as the best setup. In the other cases, we applied only length normalization without whitening. Note that we did not use any adaptation methods apart from centering on in-domain development set for all of the systems under consideration.

Table 1: *NIST 2010 det5 protocol results for pooled male and female trials*

System	Backend	EER, %	DCF10^{-3}
SpeakerMaxPoolNet7	cosine	3.65	0.587
	LDA-PLDA	3.71	0.584
	CSML	**3.45**	**0.537**
SpeakerResNet24	cosine	3.01	0.498
	LDA-PLDA	3.14	0.513
	CSML	**2.75**	**0.471**
SpeakerResNet44	cosine	2.72	0.497
	LDA-PLDA	2.76	0.526
	CSML	**2.39**	**0.464**

5.2. Training datasets

We prepared multiple training sets during our series of experiments. For preliminary ”clean” conditions studies, we used NIST's 1998-2008 datasets for training with no data augmentation.

In our main experimental setup we used telephone speech as training data. It includes Switchboard2 Phases 1, 2, and 3, Switchboard Cellular and data from NIST SREs from 2004 through 2010. In addition, we used data augmentation as it was done in [34]. Augmentation increases the amount and diversity of the training data. In total, there were about 55,000 recordings from 5,277 speakers in this training part, a major part of which is English speech. Additionally we used Russian speech subcorpus named ”RusTelecom” to extend training set. RusTelecom is a Russian speech corpus of telephone data, collected from call-centers in Russia. The training part of the RusTelecom database consists of approximately 70000 sessions from 11087 speakers.

5.3. Evaluation datasets and metrics

For ”clean” conditions studies we used the NIST 2010 evaluation dataset for testing under the det5 protocol with pooled gender trials.

Our experimental setup also includes evaluation on the Speaker-in-the-Wild [35] (SITW) and NIST SRE 2016 [36] datasets. In the case of NIST SRE 2016 we used the unequalized protocol.

We report results in terms of equal error-rate (EER) and the minimum detection cost function (DCF) with $P_{\text{Target}} = 10^{-2}$ and $P_{\text{Target}} = 10^{-3}$.

6. Results and discussion

Table 1 demonstrates speaker recognition performance obtained in ”clean” conditions only in the case of SpeakerMaxPoolNet

Table 2: *Results on NIST SRE 2016 and SITW evaluation protocols. No adaptation implemented.*

System	**Backend**	NIST2016			SITW		
		EER, %	DCF10^{-2}	DCF10^{-3}	EER, %	DCF10^{-2}	DCF10^{-3}
X-vectorNet	cosine	28.72	0.976	0.992	29.50	0.951	0.982
	LDA-PLDA	15.03	0.997	1.000	11.62	0.772	0.897
	CSML	**12.87**	**0.860**	**0.989**	**9.62**	**0.632**	**0.786**
SpeakerMaxPoolNet7	cosine	13.09	0.881	0.988	7.35	0.577	0.759
	LDA-PLDA	13.92	**0.757**	**0.901**	8.30	0.689	0.858
	CSML	**12.43**	0.837	0.972	**7.24**	**0.539**	**0.724**
SpeakerResNet24	cosine	13.94	0.894	0.987	7.08	0.535	0.703
	LDA-PLDA	14.27	**0.787**	**0.925**	7.90	0.651	0.831
	CSML	**11.79**	0.848	0.982	**6.83**	**0.510**	**0.684**

Table 3: *Results on NIST SRE 2016 and SITW evaluation protocols. Centering on in-domain devset implemented.*

System	**Backend**	NIST2016			SITW		
		EER, %	DCF10^{-2}	DCF10^{-3}	EER, %	DCF10^{-2}	DCF10^{-3}
X-vectorNet	cosine	27.56	0.950	0.976	26.10	0.989	0.997
	LDA-PLDA	12.30	0.873	1.000	11.73	0.780	0.898
	CSML	**10.45**	**0.691**	**0.924**	**8.70**	**0.622**	**0.800**
SpeakerMaxPoolNet7	cosine	11.26	0.731	0.924	6.40	0.540	0.724
	LDA-PLDA	13.51	0.736	**0.882**	7.35	0.649	0.846
	CSML	**11.09**	**0.714**	0.911	**6.64**	**0.522**	**0.711**
SpeakerResNet24	cosine	11.16	0.713	**0.902**	**5.90**	0.513	0.699
	LDA-PLDA	13.73	0.763	0.906	6.78	0.627	0.813
	CSML	**10.29**	**0.702**	0.917	6.37	**0.495**	**0.670**

and SpeakerResNet extractors corespondingly. Note that the final numbers in system names indicate the number of layers in the extractor. These results show that LDA-PLDA scoring does not improve the performance relatively to simple cosine similarity scoring. The best results we obtained were produced by proposed CSML-based systems. Another observation is that the deepest architecture SpeakerResNet with 44 layers slightly outperforms other configurations. For our futher experiments we decided to exclude so deep configuration for system training simplification. Unfortunately, at this time DNN speaker embedding systems are still unable to surpass i-vector baseline systems in terms of quality in clean conditions (see [27, 19]).

Tables 2 and 3 present the results for systems trained according to our ”in-the-wild” experimental setup. In this case we focused on all three extractors: x-vectorNet, SpeakerMaxPoolNet7, SpeakerResNet24. One can observe that cosine scoring is not suitable for x-vector based systems while SpeakerMaxPoolNet7 and SpeakerResNet24 perform well. These results are also consistent with the results obtained in the ”clean” conditions (see Table 1). The main reasons of this effect are different training strategies and optimizing loss functions. In contrast to regular softmax, angular softmax loss is specifically designed to be used with cosine similarity and trains the last network layers accordingly [8, 19].

We observe that the x-vector based LDA-PLDA system needs in-domain centering more than other studied systems (see Tables 2, 3). As shown in the tables, using CSML backend is effective and leads to the performance improvement in comparison to LDA-PLDA and simple cosine scoring.

Note that when trained with augmented data, the DNN-based speaker embedding systems significantly outperform our previous i-vector-based systems on SITW protocol [37].

7. Conclusions

A comparative study of different backend solutions for DNN-based speaker embeddings are presented in the paper. This work demonstrates that cosine similarity metric learning approach can be effectively used for speaker verification in the DNN embeddings domain. 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. Results obtained on the Speakers in the Wild and the NIST SRE 2016 evaluation sets demonstrate robustness of the CSML based systems. It is interesting to note that the proposed CSML backend model has two times fewer parameters than the PLDA model.

The successful implementation of the triplet loss CSML learning scheme for backend parameters gives grounds to hope that such strategy is suitable for whole deep speaker extractor fine tuning. This will be the topic of our further research.

8. Acknowledgements

This work was financially supported by the Ministry of Education and Science of the Russian Federation, Contract 14.578.21.0189 (ID RFMEFI57816X0189).

9. References

[1] 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.

[2] Y. Lei, N. Scheffer, L. Ferrer, and M. McLaren, "A novel scheme for speaker recognition using a phonetically-aware deep neural network," in *2014 IEEE ICASSP*, pp. 1695–1699.

[3] S. Novoselov, T. Pekhovsky, O. Kudashev, V. S. Mendelev, and A. Prudnikov, "Non-linear PLDA for i-vector speaker verification," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[4] O. Kudashev, S. Novoselov, T. Pekhovsky, K. Simonchik, and G. Lavrentyeva, "Usage of DNN in speaker recognition: advantages and problems," in *ISNN*. Springer, 2016, pp. 82–91.

[5] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[6] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich *et al.*, "Going deeper with convolutions." Cvpr, 2015.

[7] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

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

[9] E. Smirnov, A. Melnikov, S. Novoselov, E. Luckyanets, and G. Lavrentyeva, "Doppelganger mining for face representation learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1916–1923.

[10] S.-X. Zhang, Z. Chen, Y. Zhao, J. Li, and Y. Gong, "End-to-end attention based text-dependent speaker verification," in *Spoken Language Technology Workshop (SLT), 2016 IEEE*. IEEE, 2016, pp. 171–178.

[11] G. Heigold, I. Moreno, S. Bengio, and N. Shazeer, "End-to-end text-dependent speaker verification," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5115–5119.

[12] E. Variani, X. Lei, E. McDermott, I. L. Moreno, and J. Gonzalez-Dominguez, "Deep neural networks for small footprint text-dependent speaker verification," in *2014 IEEE ICASSP*. IEEE, 2014, pp. 4052–4056.

[13] S. Novoselov, O. Kudashev, V. Schemelinin, I. Kremnev, and G. Lavrentyeva, "Deep cnn based feature extractor for text-prompted speaker recognition," *arXiv preprint arXiv:1803.05307*, 2018.

[14] D. Snyder, D. Garcia-Romero, D. Povey, and S. Khudanpur, "Deep neural network embeddings for text-independent speaker verification," *Proc. Interspeech 2017*, pp. 999–1003, 2017.

[15] V. Peddinti, D. Povey, and S. Khudanpur, "A time delay neural network architecture for efficient modeling of long temporal contexts," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[16] J. Rohdin, S. Biswas, and K. Shinoda, "Discriminative plda training application-specific loss functions for speaker verification," *Odyssey*, pp. pp. 26–32, 2014.

[17] S. Prince and J. Elder, "Probabilistic linear discriminant analysis for inferences about identity," *In:ICCV*, pp. pp. 1–8, 2007.

[18] P. Kenny, "Bayesian speaker verification with heavy-tailed priors," *Odyssey*, 2010.

[19] S. Novoselov, A. Shulipa, I. Kremnev, A. Kozlov, and V. Schemelinin, "On deep speaker embeddings for text-independent speaker recognition," submitted to Odyssey proceedings, 2018.

[20] L. Zheng, K. Idrissi, C. Garcia, S. Duffner, and A. Baskurt, "Logistic similarity metric learning for face verification," *in Proc. ICASSP IEEE*, pp. 1951–1955, 2015.

[21] S. Sankaranarayanan, A. Alavi, and R. Chellappa, "Triplet similarity embedding for face verification," *CoRR*, vol. abs/1602.03418, 2016.

[22] W.-t. Yih, K. Toutanova, J. C. Platt, and C. Meek, "Learning discriminative projections for text similarity measures," *Proceedings of the Fifteenth Conference on Computational Natural Language Learning*, pp. 247–256, 2011.

[23] J. Hu, J. Lu, and Y.-P. Tan, "Discriminative deep metric learning for face verification in the wild," *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1875–1882, 2014.

[24] S. Chopra, R. Hadsell, and Y. LeCun, "Learning a similarity metric discriminatively, with application to face verification," *in Proc. CVPR IEEE*, vol. 1, pp. 539–546, 2005.

[25] D. Kedem, S. Tyree, F. Sha, G. R. Lanckriet, and K. Q. Weinberger, "Non-linear metric learning," *Advances in Neural Information Processing Systems 25*, pp. 2573–2581, 2012.

[26] H. V. Nguyen and L. Bai, "Cosine similarity metric learning for face verification," in *Proceedings of the 10th Asian Conference on Computer Vision - Volume Part II*, ser. ACCV'10. Berlin, Heidelberg: Springer-Verlag, 2011, pp. 709–720. [Online]. Available: http://dl.acm.org/citation.cfm?id=1965992.1966048

[27] 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 *Spoken Language Technology Workshop (SLT), 2016 IEEE*. IEEE, 2016, pp. 165–170.

[28] D. Povey, X. Zhang, and S. Khudanpur, "Parallel training of dnns with natural gradient and parameter averaging," *arXiv preprint arXiv:1410.7455*, 2014.

[29] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1026–1034.

[30] G. Lavrentyeva, S. Novoselov, E. Malykh, A. Kozlov, O. Kudashev, and V. Shchemelinin, "Audio replay attack detection with deep learning frameworks," *Proc. Interspeech 2017*, pp. 82–86, 2017.

[31] D. Garcia-Romero and C. Espy-Wilson, "Analysis of ivector length normalization in speaker recognition systems," *in Proc. INTERSPEECH*, pp. pp. 249–252, 2011.

[32] M. Abadi *et al.*, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org. [Online]. Available: https://www.tensorflow.org/

[33] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015, pp. 815–823.

[34] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust dnn embeddings for speaker recognition."

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

[36] N. speaker recognition evaluation 2016, "https://www.nist.gov/itl/iad/mig/speaker-recognition-evaluation-2016," 2016.

[37] O. Kudashev, S. Novoselov, K. Simonchik, and A. Kozlov, "A speaker recognition system for the sitw challenge." in *INTERSPEECH*, 2016, pp. 833–837.