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

Factorization of Discriminatively Trained i-vector Extractor for Speaker Recognition

Ondřej Novotný, Oldřich Plchot, Ondřej Glembek and Lukáš Burget

Brno University of Technology, Speech@FIT and IT4I Center of Excellence, Brno, Czechia
inovoton@fit.vutbr.cz

Abstract

In this work, we continue in our research on i-vector extractor for speaker verification (SV) and we optimize its architecture for fast and effective discriminative training. We were motivated by computational and memory requirements caused by the large number of parameters of the original generative i-vector model. Our aim is to preserve the power of the original generative model, and at the same time focus the model towards extraction of speaker-related information. We show that it is possible to represent a standard generative i-vector extractor by a model with significantly less parameters and obtain similar performance on SV tasks. We can further refine this compact model by discriminative training and obtain i-vectors that lead to better performance on various SV benchmarks representing different acoustic domains.
Index Terms: SRE

1. Introduction

In recent years, there have been many attempts to take advantage of neural networks (NNs) in speaker verification. Most of the attempts have replaced or improved one of the components of an i-vector + Probabilistic Linear Discriminant Analysis (PLDA) system (feature extraction, calculation of sufficient statistics, i-vector extraction or PLDA) with a neural network. As examples, let us mention: using NN bottleneck features instead of conventional MFCC features [1], NN acoustic models replacing Gaussian Mixture Models for extraction of sufficient statistics [2], NNs for either complementing PLDA [3, 4] or replacing it [5]. More ambitiously, NNs that take the frame level features of an utterance as input and directly produce an utterance level representation—usually referred to as an *embedding*—have in the past two years almost replaced the generative i-vector approach in text independent speaker recognition [6, 7, 8, 9, 10, 11, 12].

These embeddings are obtained by the means of *pooling mechanism*, for example taking the mean, over the frame-wise outputs of one or more layers in the NN [6], or by the use of a recurrent NN [7]. An obvious advantage—compared to i-vectors—lies in a much smaller amount of model parameters, which is typically around 10 million in the *x-vector* case [11, 12] compared to the i-vector with approximately 50 million parameters for both Universal Background Model (UBM) and i-vector extractor. This results in a very fast and memory efficient embedding extraction. A disadvantage of the x-vector framework can be seen in training during which it is essential to massively augment the training data and split them into many rather short (2–5 seconds) examples.

In this work we continue with our research from [13], where we kept the large parameter space from the generative i-vector extractor and we focused on discriminative retraining of such a model. We were able to retain the model robustness and even increase the SV performance via optimizing the model for discrimination between speakers—a task closely related to the final speaker verification. However, memory requirements and large computational cost during training have not only limited us in running experiments effectively, but more importantly it was preventing us from continuing with our research goal which is to include this model in a larger DNN scheme that is closer to an end-to-end system.

To solve our problem, we had to drastically decrease the number of trainable model parameters, but, of course, without a major decrease in performance. In the past, people have dealt with the same issue and experimented with factorization of similar or even the same models as ours. In 2003, Subspace Precision and Mean model (SPAM) for acoustic modeling in speech recognition was introduced in [14] and later optimized by Daniel Povey in [15]. SPAM models are Gaussian mixture models with a subspace constraint, where each covariance matrix is represented as a weighted sum of globally shared full-rank matrices. In 2014, Sandro Cumani proposed an i-vector extractor factorization [16], for faster i-vector extraction and smaller memory footprint, where each row of the i-vector extractor matrix is represented as a linear combination of the atoms of a common dictionary with the assumption that it is not necessary to store all rows of this matrix to perform ivector extraction.

In our approach to factorization, we were inspired by [16], but instead of factorizing each row, we perform factorization on the level submatrices of the i-vector extractor that represent individual GMM-UBM components. Also, our motivation is different, as we aim to greatly decrease the memory footprint and therefore substantially speedup the discriminative training. For now, we ignore the possible i-vector extraction speedup.

To finally obtain a discriminative i-vector extractor, we still use the same strategy as in the x-vector framework [6, 10, 11] and we retrain the NN representation of our factorized generative model to optimize the multi-class cross-entropy over a set of training speakers. This is in contrast with our previous research [17], where we optimized the binary cross-entropy over verification trials formed by pairs of i-vectors. We show that, with such an approach, we can achieve a reasonable performance. Our results are perhaps not as competitive as those achieved with current state-of-the-art x-vector systems [18], nevertheless, we are now closer to our goal which is to further use this model in the fully end-to-end discriminative system [19] that can be initialized from a robust generative baseline.

In order to compare both approaches (generative and discriminative) on a speaker verification task, both versions of i-vectors were extracted and used in a standard generative PLDA backend.

http://dx.doi.org/10.21437/Interspeech.2019-1757

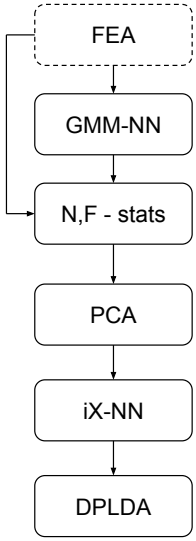

Figure 1: *Scheme of an end-to-end speaker verification system based on a feed forward NN designed to mimic a generic speaker verification system ([19]).*

2. Theoretical Background

In [19], we had built an end-to-end system (Fig. 1) that already seemingly fits our goal, but it was exactly the i-vector extractor component that posed the biggest challenge and we had to resort to ad-hoc simplifications, such as PCA-based dimensionality reduction of large dimensional sufficient statistics coming from the GMM-UBM. Our approach was to represent a standard generative i-vecror-based SV system as a series of "elementary" feed-forward NNs, each representing individual i-vector building block (e.g. GMM-UBM, i-vector extractor, PLDA classifier). In the beginning, each NN was trained separately to mimic the equivalent block from the generative training. After this "initialization", all blocks were connected and jointly retrained.

In this paper, we are focused on the i-vector extractor block and its effective discriminative retraining. We still keep the generative GMM-UBM and PLDA models.

2.1. i-vector Baseline

The i-vectors [20] provide a way of reducing large-dimensional input data to a low-dimensional feature vector while retaining most of the relevant information. The main principle is that the utterance-dependent Gaussian Mixture Model (GMM) supervector of concatenated mean vectors lies in a low-dimensional subspace—defined by a $CF \times D$ matrix $\mathbf{T} = [\mathbf{T}^{(1)\prime}, \dots, \mathbf{T}^{(C)\prime}]'$, commonly referred to as an *i-vector extractor*, with C being number of GMM components, F being feature dimensionality, and D being subspace dimensionality—and whose coordinates are given by the (D-dimensional) i-vector $\boldsymbol{\phi}$. The closed-form solution for computing the i-vector can be expressed as a function of the *zero-* and *first-order GMM statistics*: $\mathbf{n}_\mathcal{X} = [N_\mathcal{X}^{(1)}, \dots, N_\mathcal{X}^{(C)}]'$ and $\mathbf{f}_\mathcal{X} = [\mathbf{f}_\mathcal{X}^{(1)\prime}, \dots, \mathbf{f}_\mathcal{X}^{(C)\prime}]'$, where

$$N_\mathcal{X}^{(c)} = \sum_t \gamma_t^{(c)} \quad (1)$$

$$\mathbf{f}_\mathcal{X}^{(c)} = \sum_t \gamma_t^{(c)} \mathbf{o}_t, \quad (2)$$

where $\gamma_t^{(c)}$ is the posterior (or occupation) probability of frame $\mathbf{o}_t$ being generated by the mixture component c. The i-vector is then computed as

$$\boldsymbol{\phi}_\mathcal{X} = \mathbf{L}_\mathcal{X}^{-1} \bar{\mathbf{T}}' \bar{\mathbf{f}}_\mathcal{X} \quad (3)$$

with

$$\mathbf{L}_\mathcal{X} = \mathbf{I} + \sum_{c=1}^{C} N_\mathcal{X}^{(c)} \bar{\mathbf{T}}^{(c)\prime} \bar{\mathbf{T}}^{(c)}, \quad (4)$$

where $\bar{\mathbf{f}}_\mathcal{X}^{(c)}$ and $\bar{\mathbf{T}}^{(c)}$ are the "normalized" variants of $\mathbf{f}_\mathcal{X}^{(c)}$ and $\mathbf{T}^{(c)}$, respectively:

$$\bar{\mathbf{f}}_\mathcal{X}^{(c)} = \boldsymbol{\Sigma}^{(c)-\frac{1}{2}} \left(\mathbf{f}_\mathcal{X}^{(c)} - N_\mathcal{X}^{(c)} \boldsymbol{\mu}^{(c)} \right) \quad (5)$$

$$\bar{\mathbf{T}}^{(c)} = \boldsymbol{\Sigma}^{(c)-\frac{1}{2}} \mathbf{T}^{(c)}, \quad (6)$$

and $\boldsymbol{\Sigma}^{(c)-\frac{1}{2}}$ is a symmetrical decomposition (such as Cholesky) of an inverse of the GMM UBM covariance matrix $\boldsymbol{\Sigma}^{(c)}$.

2.2. Factorization of i-vector Extractor

In this work, we propose to factorize each matrix $\mathbf{T}^{(c)}$ as:

$$\bar{\mathbf{T}}^{(c)} = \sum_{q=1}^{Q} a_q^{(c)} \mathbf{U}_q, \quad (7)$$

where Q is number of factors, $\mathbf{U}_q$ are the base matrices, $a_q^{(c)}$ are scalar weights for each component $\mathbf{T}^{(c)}$. Note that bases $\mathbf{U}_q$ are shared across all components c. The number of parameters in this new model representation is $QC + QFD$, while the number parameters in the original i-vector extractor was CFD. Since there is no general requirement of linear independence for the individual matrices $\mathbf{T}^{(c)}$ in the original i-vector concept, the size of Q would have to be equal to C in order for the factorized model to fully describe the original subspace $\mathbf{T}$. However, our assumption is that there, in fact, is some level of linear dependency and therefore, Q can be chosen significantly smaller than C, therefore reducing the original model parameter space.

2.3. Discriminatively Trained Factorized i-vector Extractor

In our previous work [13], discriminative training of $\mathbf{T}$ was based on using a multi-class logistic regression with parameters $\mathbf{W}$ as a classifier (classifying K speakers), both being optimized based on the categorical cross entropy as an objective function (also depicted in Fig. 2):

$$E(\mathbf{W}, \mathbf{T}) = -\sum_{n=1}^{N} \sum_{k=1}^{K} s_{nk} \log p_\mathbf{W}(C_k \mid \boldsymbol{\phi}_{\mathcal{X}_n}), \quad (8)$$

where, s_{nk} is k-th element of the target variable in 1-of-K coding, K is the number of speakers (classes), N is the number of training samples, and $p_\mathbf{W}(C_k \mid \boldsymbol{\phi}_{\mathcal{X}_n})$ is a posterior probability (parametrized by logistic regression $\mathbf{W}$) of speaker C_k given n-th utterance. For the purpose of this work, let us treat the i-vector $\boldsymbol{\phi}_{\mathcal{X}_n}$ as a function of $\mathbf{T}$.

Generatively trained i-vector extractor was used as an initialization. In this work, we continue using this framework with

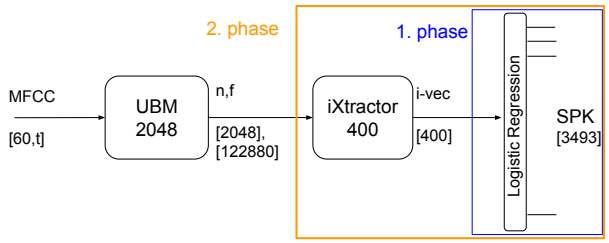

Figure 2: *Training pipeline of i-vector extractor parameters re-estimation. During the initial phase of training, only the logistic regression is trained. During the second phase, the parameters of the logistic regression and the i-vector extractor (iXtractor) are updated.*

some adjustments. Let us generalize the optimization objective by adding an L_2 regularizer:

$$E_{\text{reg}}(\mathbf{W}, \mathbf{T}) = E(\mathbf{W}, \mathbf{T}) + \lambda ||\mathbf{T}, \mathbf{T}_{\text{orig}}||, \tag{9}$$

where $||\mathbf{T}, \mathbf{T}_{\text{orig}}||$ is a Euclidian distance between our factorized matrix $\mathbf{T}$, and the original generatively trained matrix $\mathbf{T}_{\text{orig}}$.

We used two training schemes which differ in initialization and in the λ regularizing factor. In scheme-1 initialization, we select Q eigen-vectors (based on Q largest eigen-values) of covariance matrix of the vectorized $\mathbf{T}^{(c)}$'s (C vectors of FD-dimensionality). Parameters $a_q^{(c)}$ are computed as a solution of system of Q equations $\bar{\mathbf{T}}^{(c)} = \sum_{q=1}^{Q} a_q^{(c)} \mathbf{U}_q$. For this scheme, we globally set $\lambda = 0$. In phase-1 of this scheme, only classifier $\mathbf{W}$ is trained in several epochs, until convergence on a cross-validation set is reached. Then, in phase-2, both the classifier $\mathbf{W}$ and the extractor (represented by $\mathbf{U}_q$ and $a_q^{(c)}$) are retrained until convergence on a cross-validation set is reached.

In scheme-2, we started with random initialization, and for the first epoch (phase-0), λ was set to a large number (10^5 in our case). After that, λ was set to zero for the rest of the training, and phase-1 and phase-2 coppied those in scheme-1.

We experimented with different λ-progression schemes (exponential decreasing, lower stable λ during whole training, etc.), however, we discovered that one epoch was enough to reach the minimal distance to the $\mathbf{T}_{\text{orig}}$. More epochs or learning rate decreasing did not bring any significant improvement neither in $||\mathbf{T}, \mathbf{T}_{\text{orig}}||$ nor in final EERs.

In general, we used stochastic gradient descent algorithm for parameter optimization.

3. System Setup

3.1. Datasets

We used the PRISM [21] training dataset definition without added noise or reverberation to train UBM and i-vector extractor. The set comprises Fisher 1 and 2, Switchboard phase 2 and 3 and Switchboard cellphone phases 1 and 2, along with a set of Mixer speakers. This includes the 66 held out speakers from SRE10 (see Section III-B5 of [21]), and 965, 980, 485 and 310 speakers from SRE08, SRE06, SRE05 and SRE04, respectively. A total of 13,916 speakers are available in Fisher data and 1,991 in Switchboard data.

Two variants of gender-independent PLDA models were trained: one on the clean training data, the second included also artificially added different mixes of noises and reverberation. Artificially added noise and reverb segments totaled approximately 24000 segments or 30% of total number of clean segments for PLDA training, see details in Sec. 3.2.

We evaluated our systems on the *female* portions of NIST SRE 2010 [22] (*tel-tel*, *int-int* and *int-mic*) and PRISM (*prism,noi*, *prism,rev* and *prism,chn*, see section III.B of [21]), where *tel-tel* and *prism,chn* represent telephone speech, *int-int* and *int-mic* interview speech and *prism,noi* with *prism,rev* represent artificially corrupted speech with noise and reverberation.

Additionally, we used the *Core-Core* condition from the SITW challenge—*sitw-core-core*. SITW [23] dataset is a large collection of real-world data exhibiting speech from individuals across a wide array of challenging acoustic and environmental conditions.

We also test on NIST SRE 2016 [24], but we split the trial set by language into Tagalog (*sre16-tgl-f*) and Cantonese (*sre16-yue-f*). We use only female trials (both single- and multi-session). We did not use SRE'16 unlabeled development set in any way.

We randomly selected 500 utterances from 500 different speakers as a cross-validation set from the PRISM training dataset.

3.2. PLDA and i-vector Extractor Augmentation Sets

To extend the training set, we created new artificially corrupted training sets from the PRISM training set. In addition to using noise and reverberation, data were also augmented with randomly generated cuts. In our experiments, we used 30% of original training data to generate cuts with durations between 3 to 5 seconds. The composition of the augmentation set is described in details in [18].

4. Experiments and Discussion

One of the issues we had to solve to even begin experimenting with the factorized model was its proper initialization. We present two different strategies for initialization and then we will experiment with subsequent discriminative retraining of such models. We also provide comparisons with the generative baseline and with discriminative retraining of its full representation. In our experiments with factorization, we set the number of bases Q to 250. This means that the matrix $\mathbf{T}$ is represented by 7.5 times less parameters compared to the original model $\mathbf{T}_{\text{orig}}$, and when compared to the i-vector extractor block from from [19]) in Fig. 1, the number of parameters is almost half. In all of our experiments, we set the i-vector subspace dimensionality to 400.

For clarity, we denote different ways of obtaining the i-vector extractor by capital letter B, C, R and D:

B We trained a baseline i-vector extractor in the traditional generative way, using the original PRISM training corpus without any augmentations.

C_0 We initialized the bases $\mathbf{U}_b$ for factorized model by eigenvectors.

C We initialized the bases $\mathbf{U}_b$ for factorized model by eigenvectors as in C_0 and then we continued training with the loss function from (8) and the two stage training described in Sec. 2.3.

R_0 We initialized the bases $\mathbf{U}_b$ randomly and then we ran a single epoch of training with the loss function in (9).

Table 1: *Results in terms of EER* [%] *for different i-vector extractors:* B *- generative baseline without augmented data,* C_0 *and* R_0 *are mere initialized factorized models while* C *and* R *are their re-trained variants.* D *stands for a full representation of the original i-vector extractor that has been discriminatively re-trained. The table is also verticaly divided into two blocks which correspond to the training set of PLDA, where we used either only clean data or multi-condition style of training (with noised and reverberated data added to the training of PLDA).*

	PLDA clean						PLDA extension data					
Condition	B	C_0	C	R_0	R	D	B	C_0	C	R_0	R	D
tel-tel	2.23	8.39	3.9	2.47	2.2	1.97	3.36	9.72	4.91	3.52	3.3	3.25
sre16-yue-f	10.9	17.39	12.79	11.29	10.96	10.97	11.32	17.18	12.18	11.42	11.11	10.87
int-int	4.72	9.56	5.57	4.74	4.51	4.37	4.83	10.18	5.94	4.96	4.67	4.56
int-mic	2.15	5.27	2.69	2.23	2.18	2.11	2.02	5.67	2.65	2.28	2.1	1.91
prism,chn	1.13	5.63	2.25	0.92	0.83	0.88	1.14	5.95	1.98	1.11	1.12	1.14
sitw-core-core	10.51	17.97	12.35	10.92	10.4	10.29	10.57	17.54	12.33	10.84	10.47	10.21
prism,noi	4.34	11.74	6.15	4.6	4.29	3.97	3.66	10.73	5.27	4.04	3.73	3.44
prism,rev	2.81	8.59	3.67	2.84	2.49	2.54	2.45	7.25	3.17	2.49	2.3	2.34

R We initialized the bases $\mathbf{U}_b$ randomly and then we ran a single epoch of training with the loss function in (9) as in R_0, and then we continued training with the loss function from (8) and the two stage training described in Sec. 2.3.

D We driscriminatively re-trained a full representation of the baseline generative i-vector extractor [13].

To avoid over-fitting of the classifier during discriminative training, it was necessary to filter the training data. We selected speakers with at least 5 utterances in the original data. This step limits the training data to 3493 speakers with 59112 utterances (177336 utterances including augmentation).

For all experiments, we kept the same PLDA configuration. The i-vectors are pre-processed with mean normalization, LDA (i-vectors are transformed into 200-dimensions) and finally, they are length normalized.

Our results in terms of EER are presented in Tab. 1 which is divided into two vertical blocks to provide a comparison between PLDA trained on the clean data and multi-condition PLDA training, where we train the PLDA also on augmented copies of its training data. We are now interested in the general robustness of our methods and therefore we will focus on overall performance across all conditions rather than looking closely into individual cases.

The table is also divided into three horizontal blocks based on the type of the condition: into telephone channel (*tel-tel*, *sre16-yue-f*), microphone (*int-int*, *int-mic*, *prism,ch*, *sitw-core-core*) and artificially created conditions (*prism,noi*, *prism,rev*). We did not use any type of adaptation, score normalization or any other technique used for results improvement in conditions from SRE16 and others. For system C and R, we also present results for initialization, before $\mathbf{U}_b$ were retrained (in R after first epoch with $\lambda||\mathbf{T}, \mathbf{T}_{\text{orig}}||$ penalty).

When we compare baseline systems (columns B in the table) with the results obtained with initialized models for discriminative training (columns C_0 and R_0), we can see that C_0 is always significantly worse than the baseline. Initialization R_0 has much better results as they are only slightly degraded compared to the baseline indicating that we were able to represent the original i-vector model well.

We can see, that starting from C_0, we reach significant improvements with discriminative parameters re-estimation. Unfortunately, these results indicate, that the model got stuck in the local minimum and it was not able to improve to the level of the baseline.

Initialization variant R_0 proved to be a significantly better starting point. After discriminative parameter re-estimation, the model R was able to obtain slight improvement across all conditions w.r.t. R_0. Model R has also achieved a slight improvement over the baseline B or almost reached its performance.

Observing results in columns D, we can compare with discriminative retraining of the full i-vector representation [13]. With model D we achieve the best overall performance (slightly better than R), but the architecture with factorization offers approximately 4 times faster training with 7.5 times less parameters which will allow us to further extend the model and include also the GMM-UBM representation.

5. Conclusion

In this work, we have presented a way of refining a discriminative training of i-vector extractor from our previous work. We were able to slightly outperform the generative baseline. Our approach conveniently fits to the current efforts of building a fully end-to-end discriminative systems, and provides a way for a robust initialization of such a large and important part of the system. Needless to say, we have not created a new state-of-the-art system, however, we have prepared a solid platform for our further research. In our ongoing research, we will focus on the final close-form solution of the generative objective and direct estimation of $\mathbf{U}_b$, which will be helpful for simpler initialization. We also plan to analyze the effect of number of factors Q.

6. Acknowledgements

The work was supported by the Czech Ministry of Interior project No. VI20152020025 “DRAPAK”, Google Faculty Research Award program, Czech National Science Foundation (GACR) projects No. GJ17-23870Y and “NEUREM3” No. 19-26934X, and Czech Ministry of Education, Youth and Sports from the National Programme of Sustainability (NPU II) project “IT4Innovations excellence in science - LQ1602”.

7. References

[1] A. Lozano-Diez, A. Silnova, P. Matějka, O. Glembek, O. Plchot, J. Pešán, L. Burget, and J. Gonzalez-Rodriguez, “Analysis and Optimization of Bottleneck Features for Speaker Recognition,” in *Proceedings of Odyssey 2016*, vol. 2016, no. 06. International Speech Communication Association, 2016, pp.

352–357. [Online]. Available: http://www.fit.vutbr.cz/research/view_pub.php.cz.iso-8859-2?id=11219

[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 International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2014, 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 *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Sept 2015, pp. 214–218.

[4] G. Bhattacharya, J. Alam, P. Kenny, and V. Gupta, "Modelling speaker and channel variability using deep neural networks for robust speaker verification," in *2016 IEEE Spoken Language Technology Workshop, SLT 2016, San Diego, CA, USA, December 13-16*, 2016.

[5] O. Ghahabi and J. Hernando, "Deep belief networks for i-vector based speaker recognition," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2014, pp. 1700–1704.

[6] 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 International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2014, pp. 4052–4056.

[7] 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)*, March 2016, pp. 5115–5119.

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

[9] 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)*, Dec 2016, pp. 165–170.

[10] G. Bhattacharya, J. Alam, and P. Kenny, "Deep Speaker Embeddings for Short-Duration Speaker Verification," in *Interspeech 2017*, 08 2017, pp. 1517–1521.

[11] D. Snyder, D. Garcia-Romero, D. Povey, and S. Khudanpur, "Deep Neural Network Embeddings for Text-Independent Speaker Verification," in *Interspeech 2017*, Aug 2017.

[12] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust DNN Embeddings for Speaker Recognition," in *Proceedings of ICASSP*, 2018.

[13] O. Novotny, O. Plchot, O. Glembek, L. Burget, and P. Matejka, "Discriminatively re-trained i-vector extractor for speaker recognition," *accepted to ICASSP 2019*, 2019.

[14] S. Axelrod, V. Goel, B. Kingsbury, K. Visweswariah, and R. A. Gopinath, "Large vocabulary conversational speech recognition with a subspace constraint on inverse covariance matrices," in *INTERSPEECH*, 2003.

[15] D. Povey, "SPAM and full covariance for speech recognition," in *INTERSPEECH 2006 - ICSLP, Ninth International Conference on Spoken Language Processing, Pittsburgh, PA, USA, September 17-21, 2006*, 2006. [Online]. Available: http://www.isca-speech.org/archive/interspeech_2006/i06_2047.html

[16] S. Cumani and P. Laface, "Factorized sub-space estimation for fast and memory effective i-vector extraction," *Audio, Speech, and Language Processing, IEEE/ACM Transactions on*, vol. 22, pp. 248–259, 2014.

[17] O. Glembek, L. Burget, N. Brümmer, O. Plchot, and P. Matějka, "Discriminatively Trained i-vector Extractor for Speaker Verification," in *Proceedings of Interspeech 2011*, no. 8. International Speech Communication Association, 2011, pp. 137–140. [Online]. Available: http://www.fit.vutbr.cz/research/view_pub.php.cs?id=9752

[18] O. Novotný, O. Plchot, P. Matějka, L. Mošner, and O. Glembek, "On the use of X-vectors for Robust Speaker Recognition," in *Proceedings of Odyssey 2018*, no. 6. International Speech Communication Association, 2018, pp. 168–175. [Online]. Available: http://www.fit.vutbr.cz/research/view_pub.php.cs?id=11787

[19] J. Rohdin, A. Silnova, M. Diez, O. Plchot, P. Matějka, and L. Burget, "End-to-end DNN based speaker recognition inspired by i-vector and PLDA," in *Proceedings of ICASSP*. IEEE Signal Processing Society, 2018.

[20] N. Dehak, P. 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, May 2011.

[21] L. Ferrer, H. Bratt, L. Burget, H. Cernocky, O. Glembek, M. Graciarena, A. Lawson, Y. Lei, P. Matejka, O. Plchot, and N. Scheffer, "Promoting robustness for speaker modeling in the community: the PRISM evaluation set," in *Proceedings of SRE11 analysis workshop*, Atlanta, Dec. 2011.

[22] "National Institute of Standards and Technology," http://www.nist.gov/speech/tests/spk/index.htm.

[23] M. McLaren, L. Ferrer, D. Castan, and A. Lawson, "The Speakers in the Wild (SITW) Speaker Recognition Database," in *Interspeech 2016*, 2016, pp. 818–822. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2016-1129

[24] "The NIST year 2016 Speaker Recognition Evaluation Plan," https://www.nist.gov/sites/default/files/documents/2016/10/\\07/sre16_eval_plan_v1.3.pdf, 2016.