BUT System Description to SdSV Challenge 2020

Lukáš Burget, Ondřej Glembek, Alicia Lozano-Diez, Pavel Matějka, Ondřej Novotný, Oldřich Plchot, Bhargav Pulugundla, Johan Rohdin, Anna Silnova, Karel Veselý

Brno University of Technology, Speech@FIT and IT4I Center of Excellence, Brno, Czechia

{burget,lozano,isilnova,iplchot}@fit.vutbr.cz

Abstract

In this report, we describe the submission of Brno University of Technology (BUT) team to the Short Duration Speaker Verification (SdSV) Challenge 2020. For the text-dependent task, our primary submission consists of a simple linear logistic regression score level fusion of different i-vector and x-vector based systems. Our i-vector systems are based on concatenated MFCC and bottleneck features. For both types of embeddings, we use PLDA backends, showing the success of phrase-dependent training of PLDA and its combination with a Gaussian linear classifier phrase recognizer. For the task of text-independent speaker verification, we combine three different x-vector systems based on TDNN and ResNet architectures.

Index Terms: short duration speaker verification, phrase-dependent PLDA, phrase recognizer, x-vector, TDNN, ResNet

1. Task1: text-dependent

1.1. Data and Experimental Setup

To develop our systems, we used the three databases available for the challenge:

- **VoxCeleb** [1, 2]: we used the development part of VoxCeleb2 to train some of our x-vector extractors.
- **LibriSpeech** [3]: used to train some of our bottleneck feature extractors.
- **SdSV**: the in-domain data taken from the DeepMine database [4, 5]. It includes 101064 recordings of 10 different phrases (5 in English and 5 in Persian) from 963 speakers. We split this dataset into *training* and *development* sets. Our SdSV training set contains 880 speakers and 96533 utterances. Depending on the particular system, it was used to train either the bottleneck feature extractors or the embedding extractors. PLDA backends and the phrase recognizer for all the systems were trained on this SdSV training set. The cohort for score normalization (as-norm) was also created as a subset of the SdSV training data. We used 3 enrollment segments for each "speaker model" in this cohort to be consistent with the evaluation protocol.

  We set aside the other 83 speakers as our development set, which we use to create trial lists for monitoring the performance of our speaker verification systems and to train the system fusion. Our trials are created using 3 enrollment segments and do not include cross-gender trials. Out of the total 168420 trials, 4080 are target trials (i.e. target-speaker/correct-phrase (TC)), 127820 correspond to impostor/correct-phrase (IC), and the remaining 36520 are target-speaker/wrong-phrase (TW) non-target trials. We respected the proportion of wrong vs. correct phrase non-target trials declared by the challenge organizers and since they announced that majority of the wrong phrase trials would be TW, we did not include any impostor/wrong-phrase (IW) in our development set.

1.2. Utterance Embedding Extractors

We used two different x-vector extractors and four different i-vector embeddings:

1.2.1. x-vector extractors

xVoxCeleb is an x-vector extractor which is a variant of the standard Kaldi [6] TDNN model as described in [7]. This extractor is trained on VoxCeleb 16kHz audio data. The input are 40-dimensional log Mel-filter bank outputs (with frequency limits 20-7600Hz) extracted using 25 ms windows and 15 ms overlap and further normalized using short-term mean normalization with a sliding window of 3 s. The network stacks 9 TDNN layers (seeing a context of 11 frames on each side) before the pooling layer and the 512 dimensional x-vectors are extracted from the layer right after the pooling.

xSdSV is an x-vector extractor trained on the in-domain SdSV training set. In this case, we used a factorized TDNN (F-TDNN) architecture [8] trained using Kaldi but the network is trained to classify not just the speaker identities but also the phrase contained in the utterance. The features used have the same configuration as for the previous model.

1.2.2. i-vector extractors

For all our i-vector extractors, the input features are concatenated MFCCs and bottleneck features (BN). 19 MFCCs plus energy are extracted from 16kHz audio recordings using 25 ms Hamming windows with 15 ms overlap and 30 filter-bank bands. We add first and second order derivatives and discard silence frames according to an energy-based VAD (mostly skipping initial and final silence segments). Then, we apply cepstral short-term mean and variance normalization with a sliding window of 3 s. Our BN features [9] are extracted from a bottleneck layer of a neural network (NN) trained to discriminate between given phoneme units. The BN features are a frame-wise representation of the audio learned by this network. For training the NNs, we used GMM-HMM ASR models to generate the forced-alignment of the training data and this was further used either directly as the training targets or as the initial alignment for the Lattice-free MMI training [10]. We used three different variants of the BN features for the different i-vector extractors as detailed below.

We used four different i-vector extractors [11], which were all trained on the in-domain SdSV training set using an UBM-GMM with 1024 Gaussian components. The i-vector extractors only differ in the BN features used and the dimensionality of the i-vectors. The names of our i-vector systems include the dataset used to train the BN feature extractors:

iLibri800 extractor extracts 800-dimensional i-vectors. It uses the so-called *stacked BN* NN architecture [12] trained on LibriSpeech data. This architecture is composed of a cascade of two bottleneck NNs, where neighboring bottleneck-outputs from the first stage NN are stacked to define context-dependent input features for the second stage NN [9]. The NN input features are 40 log Mel-scale filter bank outputs extended with 3 kaldi-pitch features [13]. The bottleneck-outputs of the second stage NN are used as the BN features.

iLibri600 is exactly the same i-vector extractor as iLibri800 except that it produces 600-dimensional i-vectors.

iSdSV400 extracts 400-dimensional i-vectors. For BN features, it uses only the first stage NN from the stacked BN architecture described above. This BN feature extractor is trained only on the in-domain SdSV training data (i.e. only on the utterances of the 10 phrases).

iLibriSdSV400 extracts 400-dimensional i-vectors. The BN features for this system are extracted from a different architecture corresponding to the Kaldi [6] chain model, which has been modified to include the bottleneck layer[1]. This NN is trained on LibriSpeech and the in-domain SdSV challenge data together. Phonemes from LibriSpeech and SdSV data are considered as different phonemes (i.e. different classes for the NN training) although some of the SdSV sentences are in English just like LibriSpeech data.

1.3. Backends

1.3.1. Phrase-dependent PLDA (PD-PLDA)

Since both the development and evaluation data consist of only 10 phrases, all our PLDA backends were trained in a phrase-dependent fashion i.e. we train 10 different PLDA models corresponding to different phrases. Each PLDA is a two-covariance model (i.e. both within- and across-class covariance matrices are full rank). During testing, each trial is scored with the model corresponding to its enrollment phrase. Given the multi-session enrollment scenario, we use the by-the-book PLDA scoring to calculate the log likelihood verification scores. Before training or evaluating the PLDA models, the input embeddings are subject to the following pre-processing:

For our two x-vector based systems with PD-PLDA backends (systems 2 and 4 in Table 1), we center both training and evaluation x-vectors with the mean computed on the pooled data from all of the phrases from the training set. Also, a global LDA transformation reducing the dimensionality from 512 to 300 is performed, followed by a length-normalization step.

In the case of the i-vector systems, we perform phrase-dependent centering and LDA dimensionality reduction. Dimensionality after LDA is set to either 400 or 600 for different systems as indicated in Table 1 by the number appended to the backend names. Note that LDA transformation is applied even for the systems with no dimensionality reduction as it has the side effect of within-class covariance whitening, which is beneficial for the following length-normalization.

[1] `egs/librispeech/s5/local/chain/tuning/run_tdnn_1d.sh`. We removed i-vector feature adaptation and added online-cmn. The actual architecture is a Semi-Orthogonal TDNN [14]. The bottleneck is 'prefinal-l', which is the last common hidden layer preceding the split for the two objective functions in the chain model. The bottleneck has 80 dimensions, the neural network has 2×2576 outputs and 18M model parameters

1.3.2. Heavy-tailed PLDA (HTPLDA)

For some of our x-vector systems, we used a heavy-tailed PLDA (HTPLDA) [15] backend. The pre-processing of the data for HTPLDA includes centering and length-normalization. The size of the speaker subspace was set to 300 and the degrees of freedom parameter was fixed to 2. We also experimented with phrase-dependent HTPLDA backend similar to what we did with the standard PLDA. However, this approach did not outperform the results obtained with a single HTPLDA backend and was therefore not used.

1.3.3. Score normalization

To normalize the scores, we used adaptive symmetric score normalization (as-norm) which computes an average of normalized scores from z-norm and t-norm [16, 17, 18, 19]. As-norm is performed for PD-PLDA backends and it is also phrase-dependent. This means that the cohort for each phrase includes only the scores from the trials with matching enrollment phrase. For each phrase-dependent cohort we had between 618 and 779 models (enrolled from 3 utterances each). The 7011 cohort test utterances used were shared for all the phrases. Only a part of the cohort is selected to compute mean and variance for normalization and we select the 70 highest scores.

1.3.4. GLC phrase recognizer

Given that the scenario of the text-dependent task in this challenge involves a fixed set of 10 known phrases, we trained a phrase recognizer to be combined with the PLDA model outputs. This phrase recognizer is a simple Gaussian Linear Classifier (GLC) [20] trained using the i-vectors (in particular, the ones from the best single system denoted as iLibri800) on our training set. The GLC estimates the mean of each phrase and a single average within-class covariance matrix shared across the phrases.

We use this classifier in the following way: for each trial, we calculate the log-posterior probability that the test phrase contains the known enrollment phrase. Such scores have values close to zero for correct-phrase and very high negative for wrong-phrase trials. These scores are then linearly combined with the PLDA log-likelihood ratio verification scores using the logistic regression based score fusion described in Section 1.3.5.

Even though this use of the GLC phrase recognizer would not be practical in more realistic scenarios with open set of phrases, it is a good and legal approach to deal with the specific scenario of the SdSV challenge.

1.3.5. Score fusion

In order to combine the subsystems shown in Table 1 for our primary submission, we trained a linear logistic regression model to perform score level fusion. This model is trained on our development set. Thus, the results reported on that set are over-optimistic and therefore we report the results as well on the official evaluation set (from the leaderboard after the evaluation period).

1.4. Results

Table 1 summarizes the performance of the systems we built for the challenge. We show results on both the official evaluation set (obtained by submitting scores to the leaderboard for the post-evaluation phase) and our development set (comprising

Table 1: $MinDCF \times 100$ *of systems used for fusions and in final primary submission for the text-dependent task. All of them include phrase recognizer. Results on EER are not shown but followed the same trend.*

	System		Leaderboard			Development set (all trials)		
	Embedding	**Backend**	**no norm**	**as-norm**	**no+as-norm**	**no norm**	**as-norm**	**no+as-norm**
1	iLibri800	PD-PLDA400	8.61	6.31	5.87	3.4	2.35	1.82
2	xVoxCeleb	PD-PLDA300	8.15	7.65	7.35	4.59	4.12	4.01
3	iLibriSdSV400	PD-PLDA400	7.65	7.10	6.65	2.51	2.61	2.08
4	xSdSV	PD-PLDA300	11.98	9.25	9.25	6.37	4.99	4.99
5	iLibri600	PD-PLDA600	7.36	6.34	5.84	2.55	2.61	2.09
6	iSdSV400	PD-PLDA400	7.43	7.65	6.65	3.20	4.39	2.76
7	xSdSV	HTPLDA300	11.97	-	-	6.89	-	-
8	xVoxCeleb	HTPLDA300	9.20	-	-	5.00	-	-
Fusion 1+2			-	-	4.56	-	-	1.18
Fusion 1+2+3+4+ ... +8 (primary submission)			-	-	4.22	-	-	0.85
Other fusion* (leaderboard eval period)			-	-	4.09	-	-	0.79

TC, IC and TW trials). In order to effectively deal with TW trials, all these results (even for the "individual systems") used a score fusion with the phrase recognizer scores. The upper part of the table shows results for our "individual systems", while the bottom part shows score fusions of some of our systems.

We can see that as-norm proves to be effective as it helps in most of the cases. The columns denoted as *no+as-norm* correspond to a score level fusion of both original unnormalized and as-normalized scores, which often provides further significant improvements. This fusion can be seen only as a special score normalization variant and, since it uses only a single trained model (i.e. single i-vector or x-vector extractor with single PD-PLDA based backend), we consider the resulting system to be a "single system" (rather than fusion of multiple subsystems).

As *single system* we submitted our best individual i-vector system, which is the combination of no norm and as-norm scores in the first line of Table 1. In general, our i-vector based systems provide consistently better results than x-vector based systems even with the sufficient amount of training data available for the challenge. Interestingly, the xSdSV x-vector extractor trained on the in-domain SdSV training data (like our i-vector extractors) performed somewhat worse than the xVoxCeleb extractor.

Our *primary system* submitted was the fusion of all 8 individual systems shown in the penultimate line of the table. These 8 systems were selected from a larger pool of systems that we developed during the challenge, which comprises also other variants of the systems described in this document (different BN feature configurations, UBM-GMM sizes, embedding dimensionalities, x-vector extractor architectures, score normalizations, etc.). To select the subsystems, we used a greedy approach where we started from the best single system (as evaluated on our development set) and we always added one system (both as-norm and no norm scores) to the fusion that led to the biggest improvement on the development set. We also show just the fusion of two systems (1+2), one i-vector and one x-vector based (quite diverse systems), which yields 22% improvement on the evaluation set compared to the single best system. This fusion already matches the performance of the second best team in the challenge as reported in the leaderboard. The combination with a third system would already win the challenge by a significant margin. Thus, even though our primary submission was the combination of 8 systems, comparable results can be obtained by using just half of them.

The last row of the table shows results for our best performing system submitted to the challenge leaderboard prior to the deadline. This is a fusion of 11 subsystems taken from the systems pool mentioned above. However, because of its complexity, we did not select this system as our primary system.

Finally, we would like to highlight that phrase-dependent PLDA backend in combination with the phrase recognizer brought us a relative improvement on our development set of up to 63% with respect to a standard PLDA backend.

2. Task2: text-independent

Our primary submission for the text-independent speaker verification task was a fusion of three x-vector based systems. We refer to them as: xVox_TDNN_PLDA, xVoxLibriSdSV_ResNet_COS_SN and xVox_ResNet_PLDA_SN. The details of each of them are described below.

2.1. Data and Experimental Setup

Similar to our text-dependent system, we used three databases to develop our systems for the text-independent task:

- Development part of VoxCeleb2: to train all of the x-vector extractors as well as for the PLDA backend of one of the subsystems.
- LibriSpeech [3]: added to the x-vector extractor training in one of the systems.
- Development part of SdSV-task2 data. It was split into *training* and *development* sets. The training part was used to train x-vector extractors and their backend models and for score normalization. It consisted of 77239 utterances from 528 speakers. The data from the rest of the speakers (60 of them, 8525 utterances) was used to create a trial list of approximately 250k trials, where around 4k of them were target trials. Our development trial list did not include any multi-session trials.

2.2. X-vector Extractors

Our three subsystems used three different x-vector extractors:

xVox_TDNN_PLDA uses the same network as one of the extractors used for text-dependent task (xVoxCeleb system described in Section 1.2.1).

xVoxLibriSdSV_ResNet_COS_SN uses an embedding extractor based on the ResNet18 topology. It was trained on

Table 2: $MinDCF \times 100$ *and EER of systems used for fusions and in final primary submission for the text-independent task.*

	System	Leaderboard		Development set	
		$MinDCF \times 100$	EER	$MinDCF \times 100$	EER
1	xVox_TDNN_PLDA	–	–	9.85	2.62
2	xVoxLibriSdSV_ResNet_COS_SN	–	–	12.34	2.65
3	xVox_ResNet_PLDA_SN	–	–	9.81	2.27
Fusion 1+2		15.48	3.25	7.39	1.99
Fusion 1+2+3		13.17	2.71	6.85	1.85

data from 2000 speakers from VoxCeleb, Librispeech, and in-domain development data (our training part of the SdSV dataset). Large Margin Cosine Loss was used as objective with 64-dimensional Mel-filter bank outputs as the input features.

xVox_ResNet_PLDA_SN uses an extractor based on the ResNet34 topology [21]. This network uses 2-dimensional features as input and processes them using 2-dimensional CNN layers. Inspired by the x-vector topology, both mean and standard deviation are used as statistics. The network was trained on the development part of VoxCeleb2 dataset. The details of this model is given in Table 2 of [7]. However, note that for SdSV we do not do apply the additive angular margin fine-tuning.

2.3. Backends

2.3.1. Gaussian PLDA

We used Gaussian PLDA as a backend to score the embeddings in two of our subsystems. For the first one (xVox_TDNN_PLDA) we trained two PLDA models, one on in-domain training data and the other on the data from VoxCeleb dataset. Then, these two models were combined by interpolation of within- and cross-class covariance matrices of two models. Interpolation weights were set to 0.7 for SdSV model and 0.3 for VoxCeleb one. Prior to training PLDA, the data were centered (each dataset with its own mean), then the LDA dimensionality reduction was performed from 512 to 450 dimensions. LDA transformation was estimated on SdSV data for both of the models (SdSV and VoxCeleb). Finally, VoxCeleb model was a two-covariance model i.e. both speaker and channel subspace had dimensionality 450, while the in-domain model had a speaker subspace set to be 350 dimensional.

For xVox_ResNet_PLDA_SN, a single PLDA was trained on in-domain development data. The embedding preprocessing included centering and length normalization. Then, two-covariance model was trained.

Due to a large number of enroll segments in many of the multi-session trials, we average enrollment embeddings before the scoring.

2.3.2. Cosine distance

Our ResNet x-vector extractor based system denoted as xVoxLibriSdSV_ResNet_COS_SN used cosine distance scoring as backend. No preprocessing was done to the embeddings except for the centering where the mean was computed on SdSV development set. Multi-session trials, as in the PLDA case, are treated by averaging enrollment x-vectors.

2.3.3. Score normalization

Score normalization was performed for two of the subsystems: xVox_ResNet_PLDA_SN and xVoxLibriSdSV_ResNet_COS_SN. Here, as in case of text-dependent, we used adaptive symmetric snorm. The normalization cohort was formed of 3000 utterances from our SdSV training set. We select 150 highest scores to compute the mean and variance for normalization.

2.3.4. Score calibration fusion

The final submission strategy was one common fusion trained on the labeled development set created by holding out part of the Task2 training data. Each system provided scores that could be subjected to score normalization. These scores were first pre-calibrated and then passed into the fusion. The output of the fusion was then again re-calibrated.

Both calibration and fusion were trained with logistic regression optimizing the cross-entropy between the hypothesized and true labels on a development set. Our objective was to improve MinDCF error rate on the development set.

2.4. Results

The results of the individual systems as well as of the system fusion are shown in Table 2. The first two columns of the table present the results on the progress set i.e. those from the challenge leaderboard. The last two columns correspond to the results on our development set. The first three rows of the table correspond to the performance of our individual systems on our development set, and the rest shows the performance of fusions of two and three individual systems.

3. Acknowledgements

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

4. References

[1] A. Nagrani, J. S. Chung, and A. Zisserman, "Voxceleb: A large-scale speaker identification dataset," in *Interspeech 2017, 18th Annual Conference of the International Speech Communication Association, Stockholm, Sweden, August 20-24, 2017*, 2017, pp. 2616–2620. [Online]. Available: http://www.isca-speech.org/archive/Interspeech_2017/abstracts/0950.html

[2] J. S. Chung, A. Nagrani, and A. Zisserman, "Voxceleb2: Deep speaker recognition," in *Interspeech 2018, 19th Annual Conference of the International Speech Communication Association, Hyderabad, India, 2-6 September 2018.*, 2018, pp. 1086–1090. [Online]. Available: https://doi.org/10.21437/Interspeech.2018-1929

[3] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An asr corpus based on public domain audio books," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 5206–5210.

[4] H. Zeinali, H. Sameti, and T. Stafylakis, "DeepMine speech processing database: Text-dependent and independent speaker verification and speech recognition in Persian and English." in *Proc. Odyssey 2018 The Speaker and Language Recognition Workshop*, 2018, pp. 386–392.

[5] H. Zeinali, L. Burget, and J. Cernocky, "A multi purpose and large scale speech corpus in Persian and English for speaker and speech recognition: the DeepMine database," in *Proc. ASRU 2019 The 2019 IEEE Automatic Speech Recognition and Understanding Workshop*, 2019.

[6] 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*. IEEE Signal Processing Society, 2011.

[7] H. Zeinali, S. Wang, A. Silnova, P. Matějka, and O. Plchot, "BUT System Description to VoxCeleb Speaker Recognition Challenge 2019," in *VoxCeleb 2019 Workshop*, 2019.

[8] J. Villalba, N. Chen, D. Snyder, D. Garcia-Romero, A. McCree, G. Sell, J. Borgstrom, F. Richardson, S. Shon, F. Grondin, R. Dehak, L. P. García-Perera, D. Povey, P. A. Torres-Carrasquillo, S. Khudanpur, and N. Dehak, "State-of-the-Art Speaker Recognition for Telephone and Video Speech: The JHU-MIT Submission for NIST SRE18," in *Proc. Interspeech 2019*, 2019, pp. 1488–1492. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2019-2713

[9] F. Grézl, M. Karafiát, and L. Burget, "Investigation into bottleneck features for meeting speech recognition," in *Proc. Interspeech 2009*, no. 9. International Speech Communication Association, 2009, pp. 2947–2950.

[10] D. Povey, V. Peddinti, D. Galvez, P. Ghahremani, V. Manohar, X. Na, Y. Wang, and S. Khudanpur, "Purely sequence-trained neural networks for ASR based on lattice-free MMI," in *Interspeech 2016, 17th Annual Conference of the International Speech Communication Association, San Francisco, CA, USA, September 8-12, 2016*, 2016, pp. 2751–2755.

[11] 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*, 2010.

[12] F. Grézl, M. Karafiát, and K. Veselý, "Adaptation of multilingual stacked bottle-neck neural network structure for new language," in *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2014, Florence, Italy, May 4-9, 2014*, 2014, pp. 7654–7658.

[13] P. Ghahremani, B. BabaAli, D. Povey, K. Riedhammer, J. Trmal, and S. Khudanpur, "A pitch extraction algorithm tuned for automatic speech recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2014, Florence, Italy, May 4-9, 2014*, 2014, pp. 2494–2498.

[14] D. Povey, G. Cheng, Y. Wang, K. Li, H. Xu, M. Yarmohammadi, and S. Khudanpur, "Semi-orthogonal low-rank matrix factorization for deep neural networks," in *Interspeech 2018, 19th Annual Conference of the International Speech Communication Association, Hyderabad, India, 2-6 September 2018*, 2018, pp. 3743–3747.

[15] A. Silnova, N. Brummer, D. Garcia-Romero, D. Snyder, and L. s Burget, "Fast variational bayes for heavy-tailed plda applied to i-vectors and x-vectors," in *Interspeech 2018, 19th Annual Conference of the International Speech Communication Association, Hyderabad, India, 2-6 September 2018.*, 2018.

[16] P. Kenny, "Bayesian speaker verification with heavy–tailed priors," keynote presentation, Proc. of Odyssey 2010, Brno, Czech Republic, June 2010.

[17] P. Matějka, O. Novotný, O. Plchot, L. Burget, M. S. Diez, and J. Černocký, "Analysis of score normalization in multilingual speaker recognition," in *Proceedings of Interspeech 2017*. International Speech Communication Association, 2017, pp. 1567–1571. [Online]. Available: http://www.fit.vutbr.cz/research/view_pub.php.cs?id=11580

[18] D. E. Sturim and D. A. Reynolds, "Speaker adaptive cohort selection for tnorm in text-independent speaker verification," in *ICASSP*, 2005, pp. 741–744.

[19] Y. Zigel and M. Wasserblat, "How to deal with multiple-targets in speaker identification systems?" in *Proceedings of the Speaker and Language Recognition Workshop (IEEE-Odyssey 2006)*, San Juan, Puerto Rico, June 2006.

[20] D. G. Martínez, O. Plchot, L. Burget, O. Glembek, and P. Matějka, "Language recognition in ivectors space," in *Interspeech 2011*, 2011, pp. 861–864.

[21] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.