BUT Text-Dependent Speaker Verification System for SdSV Challenge 2020

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

Brno University of Technology, Faculty of Information Technology, IT4I Centre of Excellence, Czechia

{lozano, burget}@fit.vutbr.cz

Abstract

In this paper, we present the winning BUT submission for the text-dependent task of the SdSV challenge 2020. Given the large amount of training data available in this challenge, we explore successful techniques from text-independent systems in the text-dependent scenario. In particular, we trained x-vector extractors on both in-domain and out-of-domain datasets and combine them with i-vectors trained on concatenated MFCCs and bottleneck features, which have proven effective for the text-dependent scenario. Moreover, we proposed the use of phrase-dependent PLDA backend for scoring and its combination with a simple phrase recognizer, which brings up to 63% relative improvement on our development set with respect to using standard PLDA. Finally, we combine our different i-vector and x-vector based systems using a simple linear logistic regression score level fusion, which provides 28% relative improvement on the evaluation set with respect to our best single system.

Index Terms: text-dependent speaker verification, phrase-dependent PLDA, phrase recognizer

1. Introduction

In this paper, we present and analyze the text-dependent speaker verification system developed at Brno University of Technology (BUT) as the winning system in The Short-duration Speaker Verification (SdSV) challenge 2020 [1]. The SdSV challenge 2020 provides a framework for both text-dependent and text-independent scenarios. It is based on the recently released DeepMine database [2, 3], which comprises a large amount of speech recordings to train and evaluate systems using very short utterances under both scenarios. In this paper, we consider only the text-dependent speaker verification (Task 1 of the challenge), where the task is not only to verify that a test utterance contains the enrolled target-speaker voice, but also to verify the correctness of the uttered phrase, which has to match the one used for the speaker enrollment.

Recently, neural network based utterance embeddings such as x-vectors [4, 5] became popular and effective for text-independent speaker verification. For text-dependent scenarios similar to the one in the SdSV challenge, where only a small number of possible enrollment phrases are available, it was observed that it is preferable to train the embedding extractors only on the data of the matched phrases. However, before the release of the DeepMind database, the training datasets were not sufficiently large for training the data hungry x-vector extractors. An alternative approach where i-vector extractor is trained on concatenated MFCC and Bottleneck features (BN) [6], was originally proposed for text-independent speaker verification [7, 8, 9] but later found specially effective for the text-dependent task. In [10, 11], such i-vector extractor system was trained on the in-domain data with limited number of phrases and i-vectors were compared using cosine similarity as the scarce training data did not allow deployment of more conventional PLDA scoring. The SdSV challenge is the first challenge where large amounts of training data are available for the text-dependent task, which allowed us to experiment with both the x-vector based and i-vector based approaches mentioned above. For the i-vector based system we experiment with different architectures of BN features and different configurations of i-vector extractors. We also replace cosine scoring with more data hungry PLDA model. Inspired by [10, 11], where the i-vectors were normalized using phrase-dependent within-class covariance normalization, we also propose to use phrase-dependent PLDA scoring and score normalization. For x-vector based systems, we explore whether it is beneficial to train the embedding extractor on the smaller amount of available in-domain SdSV training data or the large amount of out-of-domain VoxCeleb data. To effectively reject wrong phrase utterances, we propose to use a phrase recognizer based on a Gaussian Linear Classifier (GLC) using i-vectors as input. Finally we show that linear logistic regression based score level fusion can be effectively used to combine the different systems that we developed for this challenge. Such fusion of 8 different i-vector and x-vector based systems was used to produce our SdSV challenge winning primary submission.

2. Data and Experimental Setup

The participants of the challenge were restricted to three databases to train their systems: VoxCeleb [12, 13], LibriSpeech [14], and the in-domain data taken from the DeepMine database [2, 3], which we refer to as SdSV data. We used all the databases for development of our various models. The bottleneck feature extractors were trained on LibriSpeech and the x-vector extractors were trained on the development part of VoxCeleb2. The in-domain SdSV data 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 it. 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. Finally, we also report results on the official evaluation set obtained by submitting our system outputs to the leaderboard.

3. Utterance Embedding Extractors

Our individual systems described in section 5.1 and listed in Table 1 use two different x-vectors and four different i-vectors:

3.1. x-vector extractors

xVoxCeleb is an x-vector extractor which is a variant of the standard Kaldi [15] TDNN model as described in [16]. 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 [17] 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.

3.2. 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 [6] 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 NN. 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 [18]. We used three variants of BN features for the different i-vector extractors as detailed below.

We use four different i-vector extractors [19], all trained on the in-domain SdSV training set with a UBM-GMM of 1024 Gaussian components. Our i-vector extractors only differ in the BN features used and the dimensionality of the i-vectors:

iLibri800 extractor extracts 800-dimensional i-vectors. It uses the so-called *stacked* BN NN architecture [20] trained on LibriSpeech data. This architecture is composed of a cascade of two bottleneck NNs, where neighbouring bottleneck-outputs from the first stage NN are stacked to define context-dependent input features for the second stage NN [6]. The NN input features are 40 log Mel-scale filter bank outputs extended with 3 kaldi-pitch features [21]. 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 [15] chain model, which has been, however, 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.

It should be noted that, unlike x-vector extractors names (xVoxCeleb and xSdSV), which include the data used for their training, i-vector systems names refer to the data used to train the BN feature extractor. The i-vector extractors themselves are always trained using the in-domain SdSV training data.

4. Backends

4.1. Phrase-dependent PLDA (PD-PLDA)

From the evaluation plan it was known that both the development and evaluation data consist of only 10 phrases. To take advantage of this fact, 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.

4.2. Heavy-tailed PLDA (HTPLDA)

For some of our x-vector systems, we used a heavy-tailed PLDA (HTPLDA) [23] 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] $egs/librispeech/s5/local/chain/tuning/run\_tdnn\_1d.sh$. We removed i-vector feature adaptation and added online-cmn. The architecture is a Semi-Orthogonal TDNN [22]. The bottleneck is 'prefinal-l', the last common hidden layer preceding the split for the two objective functions in the chain model. The bottleneck has 80 dimensions, the NN has 2×2576 outputs and 18M model parameters

4.3. 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 [24, 25, 26, 27]. 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.

4.4. 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) [28] 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 could use the phrase classifier to make a hard decision for each test segment and reject trials recognized as having wrong-phrase just by setting verification scores to high negative value (e.g. -inf). However, for convenience (and also to produce soft scores), we use the following procedure, which produces an outcome very similar to making hard decisions: 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 4.5.

We would like to point out that such use of GLC phrase recognizer would not be practical in more realistic scenarios with open set of phrases (even only for the wrong-phrase trials). However, this is a good and legal approach to deal with the specific scenario of the SdSV challenge.

4.5. 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 we analyze the fusion results on the official evaluation set (i.e. by submitting scores to the leaderboard).

5. Results

5.1. Individual systems

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 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 multiple systems.

Comparing *no norm* and *as-norm* columns in Table 1, we can see that as-norm proves to be effective as it helps in most of the cases. Moreover, we found that as-norm provides slightly better performance than the standard (non-adaptive) s-norm (not shown in the table). The columns denoted as *no+as-norm* correspond to a score level fusion of both original unnormalized and as-normalized scores, which interestingly often provides further significant improvements. Our preliminary experiments indicate that this fusion helps to calibrate scores for different target phrases. The fusion of the unnormalized and normalized scores 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).

Besides the *primary system* challenge participants were required to submit also a contrastive *single system*. Our submitted *single system* is the best individual system, which is the combination of no norm and as-norm scores in the first line of Table 1. This system is one of our i-vector based systems. As it can be seen from the table, 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. The organizers provided two single baseline systems, one based on x-vectors and a second one based on i-vector/HMMs as described in [1]. These systems obtained 0.5287 and 0.1464 minDCF respectively, and 9.05% and 3.49% EER, which are far from our best single system with 0.0587 minDCF and 1.89% EER.

5.2. System fusion

Our submitted *primary system* 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 work on the challenge, which comprises also other variants of the systems described in this paper (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. Table 1 also shows the results for the individual steps of this greedy fusion process. As we can see, just the fusion of two systems, one i-vector and one x-vector based, 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. Adding more systems to the fusion keeps improving performance. However, the relative improvement gradually decreases from the initial 22% relative improvement, to approximately 5% when including a third system (which would already win the challenge by a significant margin), to less than 2% when a fourth system is added. Thus, even though for our primary submission we used the combination of 8 systems, comparable results can be obtained by using just half of them. It is also interesting to observe that the systems that fuse the best are quite diverse: 1) i-vector with BN trained only on LibriSpeech, 2) x-vector trained on VoxCeleb, 3) i-vectors with BN trained also on the in-domain SdSV set, 4) x-vector trained on SdSV, and so on.

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.

Table 1: $MinDCF \times 100$ *of systems used for fusions and in final primary submission. 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			-	-	4.35	-	-	1.08
Fusion 1+2+3+4			-	-	4.28	-	-	1.02
Fusion 1+2+3+4+ ... +8 (primary submission)			-	-	4.22	-	-	0.85
Other fusion* (leaderboard eval period)			-	-	4.09	-	-	0.79

Table 2: *Comparison of PLDA, phrase-dependent PLDA (PD-PLDA), and PD-PLDA combined with the phrase recognizer (Phr-rec), over the i-vectors from our best single system (iLibri800). Results show* $minDCF \times 100$ *(EER followed the same trend).*

System		All non-target trials		Only IC non-target		Only TW non-target	
		no norm	**as-norm**	**no norm**	**as-norm**	**no norm**	**as-norm**
iLibri800	PLDA400	5.94	6.38	4.67	4.05	9.35	12.45
iLibri800	PD-PLDA400	8.27	10.26	3.64	2.59	20.4	15.48
iLibri800	PD-PLDA400 + Phr-rec	3.4	2.35	3.63	2.59	0.12	0.1

5.3. PLDA backend and phrase recognizer analysis

In this section, we analyze the PLDA backend variants when fixing the embedding extractor to the one from our best single system (iLibri800). Table 2 shows results on the development set comparing the standard PLDA backend and the phrase-dependent PLDA (PD-PLDA) backend. We report results for all trials, but also for subsets of trials where we always keep all target (TC) trials and only a specific type of non-target trials (either IC or TW). Comparing rows 1 and 2, we see that the PD-PLDA model provides significant improvements for IC non-target trials, but the performance degrades notably for TW trials. For the text-dependent task, the standard PLDA is trained with classes given by joint speaker-phrase labels. In other words, it is trained to discriminate between speaker voices and between phrases. In contrary, with PD-PLDA each phrase specific PLDA model is trained to only discriminate between speaker voices of a given phrase assuming that enrollment and test utterances contain the correct phrase. Thus, it is expected to work well for the correct-phrase trials and not for wrong-phrase trials.

To benefit from the PD-PLDA in the case of correct-phrase trials and at the same time to obtain good performance for the wrong-phrase trials, we make use of the phrase recognizer as described in Section 4.4. The results for this case are shown in the 3rd row of Table 2. We can see that the addition of the phrase recognizer does not affect the results for IC non-target trials, while the error for TW trials is reduced to almost zero. This also results in a substantial overall improvement on all the trials of the development set.

6. Conclusions

In this paper, we described the BUT winning system submitted for the text-dependent task of the SdSV challenge 2020. The amount of training data available in this challenge allowed us to investigate the commonly used text-independent speaker verification techniques for the text-dependent scenario. In particular, we studied the effectiveness x-vector extractors and compare them with i-vectors for the text-dependent scenario. While previous works on this scenario showed the importance of training embedding extractors only on data of the target phrases, we found that x-vectors trained on the out-of-domain VoxCeleb data outperform those trained on the in-domain SdSV data. However, even with the relatively large amount of in-domain training data, x-vector systems did not perform as well as the i-vector ones. Moreover, we successfully used PLDA backends in a phrase-dependent fashion, as opposed to previous works that used cosine similarity to compare embeddings. We show relative improvements of up to 63% with respect to a standard PLDA backend on our development set when we combine this phrase-dependent PLDA with a simple phrase recognizer. Finally, a simple logistic regression based score level fusion of several systems gives a 28% relative improvement over the single best system on the official challenge evaluation set.

7. Acknowledgements

The work was supported by Czech Ministry of Interior projects Nos. VI20152020025 “DRAPAK” and VI20192022169 “AI v TiV”, Czech National Science Foundation (GACR) project “NEUREM3” No. 19-26934X, European Union’s Marie Sklodowska-Curie grant agreement No. 843627, European Union’s Horizon 2020 project no. 833635 - ROXANNE and by Czech Ministry of Education, Youth and Sports from the National Programme of Sustainability (NPU II) project “IT4Innovations excellence in science” - LQ1602 and project no. LTAIN19087 “Multi-linguality in speech technologies”.

8. References

[1] H. Zeinali, K. A. Lee, J. Alam, and L. Burget, "Short-duration speaker verification (sdsv) challenge 2020: the challenge evaluation plan," 2019.

[2] 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.

[3] 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.

[4] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust dnn embeddings for speaker recognition," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018.

[5] Y. Zhu, T. Ko, D. Snyder, B. Mak, and D. Povey, "Self-attentive speaker embeddings for text-independent speaker verification," *Proc. Interspeech 2018*, pp. 3573–3577, 2018.

[6] 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.

[7] S. Yaman, J. Pelecanos, and R. Sarikaya, "Bottleneck features for speaker recognition," *Odyssey*, vol. 12, 01 2012.

[8] A. D. Lozano, 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: https://www.fit.vut.cz/research/publication/11219

[9] F. Richardson, D. A. Reynolds, and N. Dehak, "A unified deep neural network for speaker and language recognition," *CoRR*, vol. abs/1504.00923, 2015. [Online]. Available: http://arxiv.org/abs/1504.00923

[10] H. Zeinali, H. Sameti, and L. Burget, "Hmm-based phrase-independent i-vector extractor for text-dependent speaker verification," *IEEE/ACM TRANSACTIONS ON AUDIO, SPEECH AND LANGUAGE PROCESSING*, vol. 25, no. 7, pp. 1421–1435, 2017. [Online]. Available: https://www.fit.vut.cz/research/publication/11466

[11] H. Zeinali, H. Sameti, L. Burget, and J. Černocký, "Text-dependent speaker verification based on i-vectors, neural networks and hidden markov models," *Computer Speech and Language*, vol. 2017, no. 46, pp. 53–71, 2017. [Online]. Available: https://www.fit.vut.cz/research/publication/11529

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] 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

[18] 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.

[19] 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.

[20] 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.

[21] 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.

[22] 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.

[23] 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.

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

[25] 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.

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

[27] 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.

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