SPEAKER RECOGNITION FOR MULTI-SPEAKER CONVERSATIONS USING X-VECTORS

David Snyder, Daniel Garcia-Romero, Gregory Sell, Alan McCree, Daniel Povey, Sanjeev Khudanpur

Center for Language and Speech Processing & Human Language Technology Center of Excellence
The Johns Hopkins University, Baltimore, MD 21218, USA

ABSTRACT

Recently, deep neural networks that map utterances to fixed-dimensional embeddings have emerged as the state-of-the-art in speaker recognition. Our prior work introduced x-vectors, an embedding that is very effective for both speaker recognition and diarization. This paper combines our previous work and applies it to the problem of speaker recognition on multi-speaker conversations. We measure performance on Speakers in the Wild and report what we believe are the best published error rates on this dataset. Moreover, we find that diarization substantially reduces error rate when there are multiple speakers, while maintaining excellent performance on single-speaker recordings. Finally, we introduce an easily implemented method to remove the domain-sensitive threshold typically used in the clustering stage of a diarization system. The proposed method is more robust to domain shifts, and achieves similar results to those obtained using a well-tuned threshold.

Index Terms— speaker recognition, speaker diarization, deep neural networks, x-vectors

1. INTRODUCTION

Most research in speaker recognition assumes that there is only one speaker per recording and the majority of standard evaluation datasets reflect this assumption. However, speech data collected from many real-world environments violate this single-speaker assumption, and therefore benefit from speaker diarization as a preprocessing step. Speaker diarization is the process of grouping segments of speech according to the speaker, and is sometimes referred to as the "who spoke when" task. Recently, both speaker recognition and diarization have advanced significantly due to the adoption of deep neural network (DNN) embeddings to capture speaker characteristics. These embeddings are now replacing i-vectors, which have been the state-of-the-art in both tasks for almost ten years. Our work is based on x-vectors, a type of DNN embedding we developed for speaker recognition [1]. This paper studies the problem of speaker recognition for multi-speaker conversations using a modern DNN embedding-based system.

2. BACKGROUND

2.1. Speaker recognition

Until recently, most state-of-the-art speaker recognition systems were based on i-vectors [2]. The standard approach uses Gaussian mixture models (GMMs) and factor analysis to compress multiple sources of variability into a low-dimensional representation, known as an i-vector. A probabilistic linear discriminant analysis (PLDA) [3] classifier is used to compare i-vectors, and enable same-or-different speaker decisions [4, 5].

Early work using discriminatively trained neural networks to capture speaker characteristics focused on extracting frame-level features to be used as input to Gaussian speaker models [6, 7]. Heigold et al., introduced an end-to-end system, trained on the phrase "OK Google," that jointly learns an embedding along with a similarity metric to compare pairs of embeddings [8]. Snyder et al., generalized this framework to text-independent speaker recognition and inserted a temporal pooling layer into the network to handle variable-length segments [9]. The work in [1, 10] split the end-to-end approach into two parts: a DNN to produce embeddings called x-vectors, and a separately trained classifier to compare them. This facilitates use of all the accumulated backend technology developed over the years for i-vectors, such as length-normalization and PLDA scoring. The x-vector framework is described in Section 3.

2.2. Speaker diarization

Soon after their development for speaker recognition, Shum et al., adapted i-vectors to the task of speaker diarization [11, 12]. Mirroring progress in speaker recognition, recent systems have replaced i-vectors with DNN-based embeddings for capturing speaker characteristics [13, 14, 15].

A popular diarization framework involves extracting representations (i-vectors or DNN embeddings) from short speech segments, and clustering them, to discover the individual speakers in a recording. Early work used K-means or spectral clustering [11, 12]. Alternatively, a score matrix can be computed between pairs of representations using cosine distance [16] or PLDA log-likelihood ratios [17], and clustered using agglomerative hierarchical clustering (AHC) [18]. Clustering provides a coarse segmentation, which is often refined at the frame-level, using a process called Variational Bayes resegmentation [19].

2.3. Multi-speaker conversations

Capturing speaker characteristics in fixed-dimensional embeddings assumes that the input speech was generated from a single speaker, and violating this assumption reduces the effectiveness of the representation [18, 20]. Interest in the topic of speaker recognition on multi-speaker conversations has increased with the 2016 Speakers in the Wild (SITW) challenge [21] and the recent NIST 2018 Speaker Recognition Evaluation [22] due to the presence of multi-speaker enrollment and test recordings. This encourages diarization to be performed in conjunction with speaker recognition. Participants in the SITW challenge showed that diarization can significantly improve speaker recognition rates [23, 24]. Our study underscores the value of diarization for speaker recognition in the multi-speaker environment.

Table 1. *X-vector DNN architecture*

Layer	Layer Type	Context	Size
1	TDNN-ReLU	t-2:t+2	512
2	Dense-ReLU	t	512
3	TDNN-ReLU	t-2, t, t+2	512
4	Dense-ReLU	t	512
5	TDNN-ReLU	t-3, t, t+3	512
6	Dense-ReLU	t	512
7	TDNN-ReLU	t-4, t, t+4	512
8	Dense-ReLU	t	512
9	Dense-ReLU	t	512
10	Dense-ReLU	t	1500
11	Pooling (mean+stddev)	Full-seq	2x1500
12	Dense(Embedding)-ReLU		512
13	Dense-ReLU		512
14	Dense-Softmax		7185 (# spkrs)

3. X-VECTOR DNN

This section describes the x-vector DNN. The architecture is based on the DNN embedding system described in [1, 10]. Our software framework has been made available in the Kaldi toolkit [25]. An example recipe is in the main branch of Kaldi at `https://github.com/kaldi-asr/kaldi/tree/master/egs/sitw/v2` and several pretrained x-vector systems can be downloaded from `http://kaldi-asr.org/models.html`. We plan on updating the recipe and pretrained models with the improved system described in this work.

3.1. Architecture

Table 1 summarizes the architecture used in this work. The first 10 layers of the x-vector DNN consists of layers that operate on speech frames, with a small temporal context centered around the current frame t. The pooling layer receives the output of layer 10 as input, aggregates over the input segment, and computes its mean and standard deviation. These segment-level statistics are concatenated together and passed through the remaining layers of the network. The output layer computes posterior probabilities for the training speakers. Compared to the architecture described in [1], we use a slightly wider temporal context in the TDNN layers, and interleave dense layers between the TDNN layers. We found that this architecture greatly outperforms the baseline architecture available in the Kaldi recipes.

3.2. Features

The features are 30 dimensional MFCCs with a frame-length of 25 ms, mean-normalized over a sliding window of up to 3 seconds. Audio files are sampled at 16 kHz. The Kaldi energy SAD is used to filter out nonspeech frames.

3.3. Training

The DNN is trained to classify the 7,185 speakers in the training data using a multi-class cross entropy objective function. A training example consists of a 2–4 second speech segment (about 3 seconds average), along with the corresponding speaker label. Following a study by McLaren et al. in [26], we use much more aggressive data augmentation than in previous studies (see Section 6.1), train the DNN for 6 epochs (instead of 3) and use a minibatch size of 128 (instead of 64).

3.4. Embedding extraction

Once the network is trained, x-vectors are extracted from the affine component of layer 12. The x-vectors are used as features for two different PLDA backends (one for the diarization system described in Section 4 and one for the speaker recognition system described in Section 5).

4. SPEAKER DIARIZATION

The diarization system is based on a system we developed for the 2018 DIHARD speaker recognition challenge [14, 27]. A similar recipe (for narrowband telephone speech) can be found in the main branch of the Kaldi toolkit: `https://github.com/kaldi-asr/kaldi/tree/master/egs/callhome_diarization/v2`. The system uses x-vectors extracted from the DNN in Section 3 with PLDA, and agglomerative hierarchical clustering (AHC). The PLDA backend consists of centering, whitening and length normalization, followed by scoring. All components of the backend are trained on 3 second segments extracted from the augmented VoxCeleb data described in Section 6.1.

For either an enrollment recording or a test recording, x-vectors are extracted from 1.5 second segments with a 0.75 second overlap. PLDA scores are computed between all pairs of x-vectors. This is followed by AHC with average linkage clustering. In our primary system, the number of clusters is controlled by a stopping threshold which was tuned on the held-out SITW *DEV* set. The most similar clusters are repeatedly merged, until the average PLDA scores between clusters is less than the threshold. Diarization results in N clusters (which, ideally correspond to speakers).

4.1. Removing the AHC threshold

AHC-based diarization typically requires a well-chosen cluster stopping threshold to achieve good performance. This threshold is sensitive to the domain of the data, and a poorly chosen threshold will result in bad performance. This is a particularly concerning possibility when a reliable development set is not available.

To improve robustness, we propose a simple alternative to eliminate the need for the AHC threshold. Instead of relying on a tuned AHC threshold, we begin with an estimate of the maximum number of speakers K that might appear in the recordings. We assume that there are never more than K speakers in an utterance, and perform clustering K times, with exactly $k \in \{1, 2, \ldots, K\}$ clusters each time we perform clustering. Taking the union of each of the individual diarizations results in a set of $N = \frac{K(K+1)}{2}$ ways to partition a recording that has at most K speakers. The N potential speakers are then treated exactly the same as the speakers discovered by clustering with an AHC threshold, as described in Section 5.

Looking at the SITW *DEV* set, we found that the performance isn't very sensitive to different values of $K \geq 3$. We use $K = 5$ for the experiments in the results section.

4.2. Diarizing enrollment recordings

If we are processing an enrollment recording, then the goal is to use an assist segment to identify any other speech in the recording which belongs to the speaker we wish to enroll, while removing any speech

belonging to other speakers. As described in Section 6.2, an assist segment is about 5 seconds of speech in a longer recording, which is known to contain the speaker we wish to enroll.

The speech corresponding to the assist segment is treated as an "auxiliary enrollment" and the entire recording is treated as an "auxiliary test" recording. After clustering, we obtain N speakers in the auxiliary test. We then perform the procedure described in Section 5, which involves computing PLDA scores between the auxiliary enrollment and each of the N speakers discovered in the auxiliary test. All the speech segments belonging to the speaker in the auxiliary test that maximizes the PLDA score (as in Equation 1) are identified, and used by the speaker recognition system to extract an enrollment x-vector.

4.3. Diarizing test recordings

Handling the test recordings is straightforward once AHC is performed. The speech segments are grouped according to the N speakers discovered in the conversation, and are passed directly to the speaker recognition system, where they are used to perform recognition as described in the next section.

5. SPEAKER RECOGNITION

Recognition is performed using x-vectors extracted from the DNN in Section 3 and a PLDA backend. The x-vectors are centered, dimensionality reduced to 225 using LDA, and are length-normalized. All parameters in the backend are estimated on the augmented VoxCeleb data, as described in Section 6.1.

If diarization was performed on a test recording, then, instead of extracting a single x-vector for the entire test recording, we extract N x-vectors, one for each of the N speakers identified in the recording. Suppose $R(,)$ is the PLDA log-likelihood ratio score, $\mathbf{u}$ is the x-vector for the enrolled speaker and $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_N$ are the x-vectors for each of the N speakers in the test recording. To perform speaker recognition, we compute the PLDA score as in Equation 1, which is the maximum of the PLDA scores between the enrollment x-vector and all N test x-vectors.

$$R(\text{enroll}, \text{test}) = \max\{R(\mathbf{u}, \mathbf{v}_1), \ldots, R(\mathbf{u}, \mathbf{v}_N)\} \qquad (1)$$

Handling a diarized enrollment recording is simpler, since there can only be one speaker of interest at a time. We simply extract the enrollment x-vector from all speech frames identified as belonging to the speaker of interest (as described in Section 4.2), and ignore the remaining frames.

6. EXPERIMENTAL SETUP

6.1. Training data

The system is trained on a large subset of the combined VoxCeleb 1 [28] and VoxCeleb 2 [29] corpora sampled at 16 kHz. The test portion of VoxCeleb 2 as well as 60 speakers from VoxCeleb 1 overlap with the evaluation dataset, and so we removed them before training. See `http://www.openslr.org/resources/49/voxceleb1_sitw_overlap.txt` for a list of speakers from VoxCeleb 1 which are known to overlap with SITW. This leaves a total of over 150,000 recordings from 7,185 speakers. Using the target speaker marks provided in the corpora, the recordings are split into over 1.2 million segments.

We apply a data augmentation strategy based on [1] that consists of adding noises, music, babble, and reverberation. The x-vector DNN was trained on 7.2 million segments, comprised of the 1.2 million "raw" segments extracted directly from VoxCeleb, plus an additional 6 million segments obtained by data augmentation. The PLDA backend for speaker recognition (Section 5) was trained on the full-length recordings of VoxCeleb, but we only keep the speech belonging to the speakers of interest (as provided by the segments that are distributed with the corpora). We apply augmentation to double the amount of training data, which increases the number of recordings from about 150,000 to 300,000. Finally, the diarization backend (Section 4) was trained on 256,000 three second segments extracted randomly from the full-length augmented recordings.

6.2. Speakers in the Wild

We perform experiments on the Speakers in the Wild (SITW) dataset developed by SRI International [21]. The dataset consists of challenging audio collected from diverse conditions in the video audio domain. One of the challenges is the presence of multiple speakers in some of the utterances. The recordings vary in length, from 6 to 240 seconds.

The dataset is divided into a development set *DEV* (which we use only for tuning) and an evaluation set *EVAL*. The *EVAL* set contains 180 speakers divided into 4,170 models and a total of 2,883 audio files.

Enrollment conditions

- *CORE:* Enrollment recordings contain exactly one speaker.
- *ASSIST:* One or more speakers in enroll, along with an "assist" mark, which is a short segment (typically 5 seconds) of the recording that is known to contain the speaker of interest.

Test conditions

- *CORE:* Test recordings contain exactly one speaker.
- *MULTI:* One or more speakers in the test recordings.

7. EXPERIMENTAL RESULTS

In Table 2 we report results on the *EVAL* portion of the Speakers in the Wild (SITW) dataset. The four evaluation conditions are formed by pairing an enrollment condition with a test condition described in Section 6.2. Performance on these conditions is examined in Sections 7.1–7.4. The results are further broken down by whether or not the enroll or test recordings are diarized. The diarization system and its interaction with speaker recognition is the subject of Sections 4–5. We report results in terms of equal error rate (EER) and the minimum of the normalized detection cost function (DCF). DCF1 uses P_{Target}=10^{-2} and DCF2 uses P_{Target}=10^{-3}.

The *Threshold* system uses an AHC threshold tuned on the *DEV* set to control the number of speakers, whereas *No threshold* uses the alternative method described in Section 4.1 to eliminate the threshold. In Section 7.5, we discuss performance using the proposed alternative system that eliminates the AHC threshold.

7.1. CORE-CORE

In the simplest SITW evaluation condition, there is exactly one speaker present in both the test and enrollment recordings. In the first row of results in Table 2 (NO DIAR), we do not apply any diarization and achieve very low error rates. In the next row of results (TEST), we apply diarization to the test recordings. Using the standard approach, diarizing single-speaker recordings degrades performance by a very small amount–less than half a percent relative on all performance metrics.

Table 2. *Results on the SITW evaluation set.*

		EVAL CORE-CORE			EVAL CORE-MULTI			EVAL ASSIST-CORE			EVAL ASSIST-MULTI		
	Diarization	**EER**	**DCF1**	**DCF2**	**EER**	**DCF1**	**DCF2**	**EER**	**DCF1**	**DCF2**	**EER**	**DCF1**	**DCF2**
	NO DIAR	1.7	0.20	0.34	3.5	0.28	0.44	3.2	0.24	0.38	4.3	0.28	0.43
Threshold	ENROLL	1.8	0.21	0.35	2.1	0.22	0.41	1.6	0.20	0.35	3.0	0.26	0.41
	TEST							3.3	0.24	0.39	3.8	0.26	0.41
	BOTH							1.7	0.21	0.36	2.1	0.21	0.37
No threshold	ENROLL	1.8	0.23	0.36	2.0	0.22	0.40	1.6	0.20	0.36	3.0	0.26	0.42
	TEST							3.8	0.26	0.40	3.9	0.26	0.41
	BOTH							2.2	0.23	0.38	2.2	0.22	0.38

CORE-CORE is the most commonly used condition from SITW. Our best performance on this condition is EER=1.7% DCF1=0.20, which comfortably outperforms the best previously reported numbers in [30], which are EER=2.7% and DCF1=0.33. The x-vector DNN architecture in this paper is similar to that of the previous work, so the improvements are mostly due to a better training recipe, which consists of more aggressive data augmentation than previously used, and the addition of a substantial amount of in-domain data from the VoxCeleb 2 Corpus [29].

7.2. CORE-MULTI

CORE-MULTI extends the previous condition with test recordings that contain one or more speakers. We still use single-speaker enrollment recordings in this condition.

Diarizing the multi-speaker test conversations (TEST) results in a clear improvement over performing no diarization (NO DIAR). Using a tuned AHC threshold, diarization reduces EER by 38%, and by 20% in DCF1 and 8% in DCF2. The results that eliminate the AHC threshold are even slightly better. Note that we do not consider the effect of diarizing the enrollment recordings yet, as we do not consider that meaningful unless the assist segments are provided.

7.3. ASSIST-CORE

This condition introduces our systems to the assist segments. These segments provide a few seconds of speech of the speaker we wish to enroll. As described in Section 4.2, we use the assist marks to discover additional speech (in the enrollment recording) that belongs to the speaker of interest, while discarding any speech from other speakers. Although the enrollment recordings may have multiple speakers, the test recordings are single-speaker in this condition.

Diarizing the enrollment recordings (ENROLL) reduces EER by 50% relative to NO DIAR. The DCF numbers also improve, but by a smaller amount. As expected, unnecessarily diarizing the test recordings (but not enrollment) results in the worst performance. Nonetheless, the *Threshold* results are not significantly worse than the results without diarization. In the last row (BOTH), we diarize both the enrollment and the test recordings. For the *Threshold* system, this degrades performance by 2–8% relative to the ENROLL results, but still maintains an improvement over NO DIAR.

7.4. ASSIST-MULTI

This condition combines the challenge of potential multi-speaker enrollment recordings with multi-speaker test recordings. As in the previous section, diarizing the enrollment recordings is enabled by the assist segments.

Diarizing either enroll or test recordings individually (but not together) results in moderate improvements in EER, and smaller improvements in DCF1 and DCF2. Fortunately, the benefit of combining enroll and test diarization results in much more dramatic improvements. Looking at the *Threshold* system, we observe a 50% EER reduction over no diarization and a 14–23% reduction in DCF.

7.5. Removing the threshold

The previous sections showed that the *Threshold* system achieves excellent results. It relies on an AHC threshold tuned on labeled in-domain data. Although this is not an obstacle for this paper, as we are able to tune on the well-matched *DEV* set, it cannot be assumed that an in-domain development set is always available. The *No threshold* system uses the method described in Section 4.1 to address the problem of performing diarizing when no development set is available to tune on.

In Table 2 we see that, under most conditions, the alternative *No threshold* system performs similarly to *Threshold*. When diarizing is required for multi-speaker conversations, the results of this system are very similar to the standard approach. The system performs worst on ASSIST-CORE when we needlessly diarize the test recordings. However, the BOTH results are nonetheless better than the results without diarization.

8. CONCLUSIONS

This paper investigated speaker recognition with multi-speaker recordings. We used a diarization system based on x-vectors, PLDA, and agglomerative hierarchical clustering (AHC) as a front-end for a speaker recognition system. We evaluated performance on the Speakers in the Wild dataset, and found that diarization significantly improved speaker recognition performance on multi-speaker conversations, and retained strong performance on single-speaker recordings as well. Finally, we showed that the AHC threshold, which controls the number of clusters, can be replaced with an alternative method that achieves similar performance under most conditions, but eliminates the need for a in-domain development set for tuning.

9. ACKNOWLEDGMENTS

This material is based upon work supported by the National Science Foundation Graduate Research Fellowship under Grant No. 1232825. Any opinion, findings, and conclusions or recommendations expressed in this material are those of the authors(s) and do not necessarily reflect the views of the National Science Foundation.

10. REFERENCES

[1] 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.

[2] 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, 2011.

[3] S. Ioffe, "Probabilistic linear discriminant analysis," *Computer Vision–ECCV 2006*, pp. 531–542, 2006.

[4] P. Kenny, "Bayesian speaker verification with heavy-tailed priors.," in *Odyssey*, 2010, p. 14.

[5] N. Brümmer and E. De Villiers, "The speaker partitioning problem.," in *Odyssey*, 2010, p. 34.

[6] L. Heck, Y. Konig, K. Sonmez, and M. Weintraub, "Robustness to telephone handset distortion in speaker recognition by discriminative feature design," in *Speech Communication*, 2000, vol. 31, pp. 181–192.

[7] A. Salman, *Learning speaker-specific characteristics with deep neural architecture*, Ph.D. thesis, University of Manchester, 2012.

[8] 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)*. IEEE, 2016, pp. 5115–5119.

[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 *Spoken Language Technology Workshop (SLT)*. IEEE, 2016.

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

[11] S. Shum, N. Dehak, E. Chuangsuwanich, D. Reynolds, and J. Glass, "Exploiting intra-conversation variability for speaker diarization," in *Twelfth Annual Conference of the International Speech Communication Association*, 2011.

[12] S. Shum, N. Dehak, and J. Glass, "On the use of spectral and iterative methods for speaker diarization," in *Thirteenth Annual Conference of the International Speech Communication Association*, 2012.

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

[14] G. Sell, D. Snyder, A. Mccree, D. Garcia-Romero, J. Villalba, M. Maciejewski, V. Manohar, N. Dehak, D. Povey, S. Watanabe, and S. Khudanpur, "Diarization is Hard: Some Experiences and Lessons Learned for the JHU Team in the Inaugural DIHARD Challenge," in *Proceedings of the 19th Annual Conference of the International Speech Communication Association, INTERSPEECH 2018*, Hyderabad, India, sep 2018, pp. 2808—-2812.

[15] Q. Wang, C. Downey, L. Wan, P. Mansfield, and I. Lopez Moreno, "Speaker diarization with lstm," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5239–5243.

[16] M. Senoussaoui, P. Kenny, T. Stafylakis, and P. Dumouchel, "A study of the cosine distance-based mean shift for telephone speech diarization," *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 22, no. 1, pp. 217–227, 2014.

[17] G. Sell and D. Garcia-Romero, "Speaker diarization with plda i-vector scoring and unsupervised calibration," in *Spoken Language Technology Workshop (SLT), 2014 IEEE*. IEEE, 2014, pp. 413–417.

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

[19] M. Diez, L. Burget, and P. Matejka, "Speaker diarization based on bayesian hmm with eigenvoice priors," in *Proc. Odyssey 2018 The Speaker and Language Recognition Workshop*, 2018, pp. 147–154.

[20] A. Martin and M. Przybocki, "Speaker recognition in a multi-speaker environment," in *Seventh European Conference on Speech Communication and Technology*, 2001.

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

[22] "NIST speaker recognition evaluation 2018," `https://www.nist.gov/sites/default/files/documents/2018/08/17/sre18_eval_plan_2018-05-31_v6.pdf`, 2018.

[23] O. Novotnỳ, P. Matejka, O. Plchot, O. Glembek, L. Burget, and J. Cernockỳ, "Analysis of speaker recognition systems in realistic scenarios of the sitw 2016 challenge.," in *Interspeech*, 2016, pp. 828–832.

[24] Y. Liu, Y. Tian, L. He, and J. Liu, "Investigating various diarization algorithms for speaker in the wild (sitw) speaker recognition challenge.," in *Interspeech*, 2016, pp. 853–857.

[25] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlíček, Y. Qian, P. Schwarz, et al., "The Kaldi speech recognition toolkit," in *Proceedings of the Automatic Speech Recognition & Understanding (ASRU) Workshop*, 2011.

[26] M. McLaren, D. Castan, M. Nandwana, L. Ferrer, and E. Yılmaz, "How to train your speaker embeddings extractor," in *Odyssey: The Speaker and Language Recognition Workshop, Les Sables dOlonne*, 2018.

[27] N. Ryant, K. Church, C. Cieri, A. Cristia, J. Du, S. Ganapathy, and M. Liberman, "First dihard challenge evaluation plan," 2018.

[28] A. Nagrani, J. S. Chung, and A. Zisserman, "Voxceleb: a large-scale speaker identification dataset," in *Interspeech*, 2017.

[29] J. S. Chung, A. Nagrani, and A. Zisserman, "Voxceleb2: Deep speaker recognition," in *INTERSPEECH*, 2018.

[30] A. Silnova, N. Brümmer, D. Garcia-Romero, D. Snyder, and L. Burget, "Fast Variational Bayes for Heavy-tailed PLDA Applied to i-vectors and x-vectors," in *Interspeech 2018*, Hyderabad, India, 2018, pp. 72–76.