ISSN 1870-4069

DNN Speaker Tracking with Embeddings

Carlos Rodrigo Castillo-Sanchez[1], Leibny Paola Garcia-Perera[2], Anabel Martin-Gonzalez[1]

[1] Universidad Autónoma de Yucatán, Computational Learning and Imaging Research, Mexico

[2] Johns Hopkins University, Center for Language and Speech Processing, USA

carloscastillomvc@gmail.com, amarting@correo.uady.mx, leibny@gmail.com

Abstract. Speaker tracking is the task of finding hypothesized speakers in a multi-speaker conversation. In this paper, we propose a novel way to perform online speaker tracking based on neural networks. We designed an architecture that mimics the probabilistic linear discriminant analysis (PLDA) algorithm and outputs the most likely regions uttered by a predefined target speaker. This output can be used for downstream tasks such as diarization or tracking, as analyzed in this paper. For sake of generalization, we used two standard public datasets that were carefully modified to create two-speaker subsets with additional overlapping speech and non-target speakers. Relative improvements of 40% and 20% in minDCF for CALLHOME and DIHARD II single-channel show promising performance.

Keywords: Speaker tracking, speaker diarization, speaker verification, x-vector, i-vector.

1 Introduction

Speaker tracking can be considered as the process of identifying all regions uttered by a hypothesized speaker in a multi-speaker recording [1]. Similarly to speaker diarization, which answers the question *"who spoke when?"*, speaker tracking searches for those regions, but assigns speaker identities to them. Finding where a given speaker is intervening in a conversation is an essential preprocessing step for many multi-speaker applications, where speech data from previous enrollments may be available, such as virtual assistants, meetings and broadcast news transcription and indexing [8].

As shown in [6], diarization and tracking are two methods closely related. Although tracking would benefit from the diarization, in this research, we explored the possibility of including a neural network as a robust classifier that

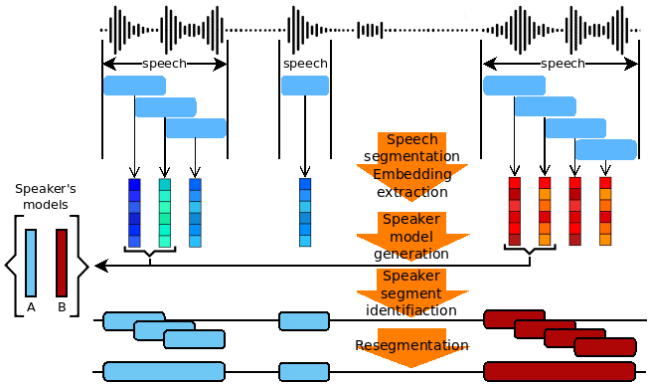

Fig. 1. Pipeline of the proposed speaker tracking system.

can operate similarly to the probabilistic linear discriminant analysis (PLDA), with the goal of naturally providing results for diarization and tracking.

Since there are just a few studies on speaker tracking [1,8,24], we use diarization as the main background and inspiration of this work. Most of the standard speaker diarization systems focus on offline clustering as it uses all the contextual information to label the speech regions. Examples of such algorithms include agglomerative hierarchical clustering (AHC) [10,13], k-means [14,2] and spectral clustering [11,15]. These clustering methods cannot be used in real-time applications since they require complete speech data upfront. Latency-sensitive applications must have speaker labels generated as soon as speech segments are available to the system. We reviewed diarization approaches that are effective in an online setup. In [27] an embedding-based speaker diarization system is presented, it uses *d-vectors* [26] with an LSTM-based scoring function in combination with spectral clustering to successfully perform offline diarization; however, the diarization error rate almost doubles in its online modality. Another online diarization approach is introduced in [7], they propose a deep neural network (DNN) embedding suitable for online processing referred to as speaker-corrupted embedding. The diarization algorithm uses cosine similarity to compare the speaker models and the segments embeddings to make the labeling decisions. A promising approach for diarization is the use of acoustic features of a speaker to target the system's detection to their speech. In [12], an initial estimation of target's speaker features (i-vectors) is performed with clustering-based diarization, providing excellent performance in CHiME-6. Although, this is an offline approach, it could be extended to an online setup.

In this paper, we propose an online speaker tracking pipeline by replacing the unsupervised offline clustering module from the standard diarization system with an online tracking method that uses a DNN as a robust embedding classifier. The main idea is to mimic the PLDA, scoring the similarity of each hypothesized speaker at every segment of a recording. As shown in Fig. 1, our speaker tracking

system shares many of its components with the standard diarization pipeline (segmentation, embedding extraction, clustering, and resegmentation) [4,30,18], with the main difference being the removal of the clustering algorithm.

The experimental results on CALLHOME and DIHARD II single-channel [16] reveal that our method achieves competitive results in comparison to the PLDA baseline, while improving the verification performance in EER and minDCF[3].

2 Methodology

In this section, we introduce our speaker tracking framework; Fig. 1 illustrates the overall steps of our tracking pipeline.

2.1 Speech Segmentation and Embedding Extraction

The first module in our pipeline is inspired by the standard diarization system. It uses a Voice Activity Detector (VAD) to determine the speech parts in the input audio signal, excluding the non-speech regions from subsequent processing. A sliding window further divides these regions into a set of smaller, overlapping speech segments, which are the units of audio that can be attributed to a speaker, establishing the temporal resolution of the speaker tracking results. We decided to use an oracle VAD as a segmentation mechanism to focus our efforts on checking whether our proposed architecture can track speakers accurately.

Embedding extraction The next step in the pipeline is to extract an embedding from each segment; such embeddings will be used in two tasks: develop the hypothesized speaker's models and label the segments. Our system was tested following the i-vector- and x-vector-based approaches [17,20]. The i-vector, introduced by Dehak *et al.* [3], is a speaker representation that provides a way to reduce large-dimensional input speech data to a small-dimensional feature vector that retains most of the relevant channel and speaker information. The x-vector, introduced by Snyder *et al.* [23,20] is an embedding extracted from a deep neural network trained to discriminate between speakers, mapping variable-length speech segments to a fixed-length feature vector. Nowadays, the x-vector approach provides state-of-the-art performance in many speaker recognition fields, such as speaker verification and speaker diarization [19,22,28,16].

2.2 Speaker Model Generation

After extracting the segment embeddings, a speaker model is generated for each hypothesized speaker. In our experimental setup, we compute each speaker model by averaging its first embeddings from ground truth labels. The number of embeddings used in this process depends on a tunable time window that will be

[3] Code available at: https://github.com/CarlosRCS9/kaldi/tree/paper-dnn-tracking/egs/dnn_tracking/v1

analyzed later in this research. We define the variable *model time* as the window width used to generate the speakers' models.

With this approach, the system operates in an online fashion in which, with a few labeled samples of the target speakers, it can find their appearances along the complete audio. In a real-life scenario, we expect to have speech data from the target speakers from previous enrollments or a method to record a speaker model, such as a calibration procedure.

2.3 Speaker Segment Identification

The resulting segment embeddings and the speakers' models are then passed through a speaker identification/verification stage. The speaker-tracking DNN, the key component of our pipeline, performs this task.

According to the run-time latency, the speaker identification module follows an **online** tracking strategy. It produces a speaker label immediately after a segment is available without the knowledge of future segments, making it easier for the system to deal with large amounts of audio data since the clustering stage is no longer used.

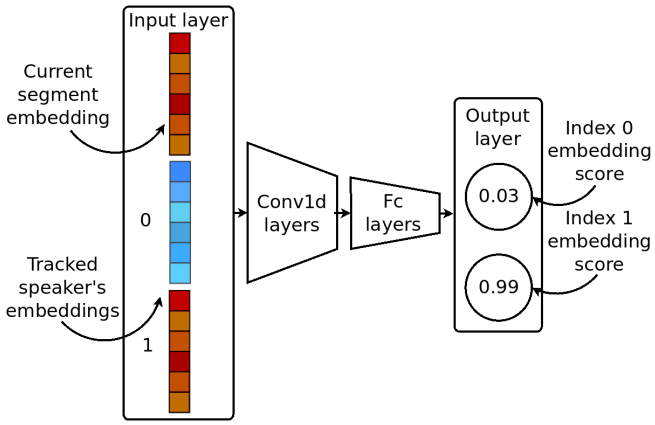

Fig. 2. Network input and output layer for the segment identification process.

Features Fig. 2 illustrates the structure of the network's input and output layers during the segment labeling process. For a given utterance, the input and output sequences of the network (X', Y') are defined as follows:

– The speech segmentation and embedding extraction module provides a sequence of embeddings $X = (x_1, x_2, \ldots, x_T)$, where each $x_t \in \mathbb{R}^b$ has a 1:1 correspondence to the T segments obtained from the input utterance, and b is the dimension of every embedding.

- The speaker model generation module provides the sequence $M = (m_1, m_2, \ldots, m_S)$ where $m_s \in \mathbb{R}^b$, such that each entry of the sequence is a model of one of the S tracked speakers.
- The input sequence of our network is defined as the concatenation of M to each element of X. $X' = \{x_t^\frown M | x_t \in X\}$.
- The sequence $Y = (y_1, y_2, \ldots, y_T)$ is given by the speaker labels of the T segments.
- The output sequence is given by $Y' = \{\Phi(y_t) | y_t \in Y\}$ where $\Phi(y_t) = \{P(m_s|x_t, y_t) | m_s \in M\}$. At training time, Y is given by the ground-truth labels. At inference, Y is computed by the estimated labels.

Architecture Table 1 summarizes the final DNN architecture used in this work. The first three convolutional layers of the network provide a comparison stream for each of the S speakers models and the current audio segment. The similarity measure between the segment embedding and the input speaker models is hence computed using the contextual information of all the speaker models. Note that our architecture intends to track up to S speakers simultaneously. To track less than S speakers, it is required to add zero-padding in the input layer at the location where a speaker model would be.

The last fully-connected feed-forward layers use the S comparison streams to score the similarity of the target speaker model and the incoming segment, with the last layer having a sigmoid activation function instead of softmax. Such activation function allows the network to provide zero scores in all of its outputs when a segment does not belong to any of the tracked speakers, as shown in Fig. 3.

Table 1. Speaker-tracking DNN architecture.

Layer type	**Filters**	**Kernel**	**Input × output**
Conv1d.ReLU	S^3	3	$b(S+1) \times (b-2)S^3$
Conv1d.ReLU	S^2	3	$(b-2)S^3 \times (b-4)S^2$
Conv1d.ReLU	S	3	$(b-4)S^2 \times (b-6)S$
Dense.ReLU			$(b-6)S \times 32S$
Dense.ReLU			$32S \times 16S$
Dense.Sigmoid			$16S \times S$

Training During training, all possible permutations of the elements of M are computed and appended to every input x_t with two main goals: reduce overfitting by forcing all output neurons to score the same speaker models, and augment the number of training samples. This procedure ensures the DNN scoring to be independent of the speaker model permutation order. Fig. 3 shows how the training data is furthermore augmented by adding zero-padding as a non-speaker

model feature. This procedure simulates a verification task during training since the network has to decide whether the current segment embedding belongs to one of the available models or not.

At inference time, our system initializes with an array of hypothesized speaker models (with length less or equal to S). With each recording segment, the similarity of each hypothesized speaker is computed. This is done by appending the models' array to the segment embedding as the network's input, with the output neurons providing similarity scores for each speaker. In an identification setup, we label the segment with the highest score index. If the task requires verification, a certainty threshold is used to label the segments.

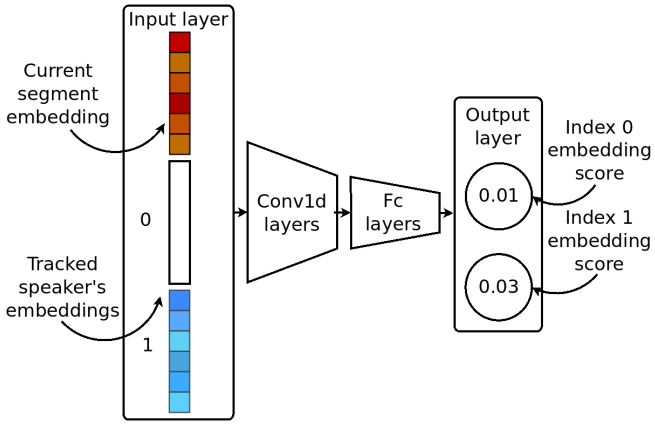

Fig. 3. Input layer with zero padding.

2.4 Probabilistic Linear Discriminant Analysis (PLDA)

The baseline system uses probabilistic linear discriminant analysis (PLDA) scoring as the similarity measure[4]. It has been proven to achieve state-of-the-art performance in many speaker recognition tasks. It provides a powerful distortion-resistant mechanism to distinguish between different speakers and robustness to same variability [9,31,16,29].

2.5 Post-processing

Due to the **online** nature of our pipeline, the post-processing step is applied as soon as a speaker label is inferred. This step refines the tracking results by performing two tasks: merging the contiguous segments that share the same label, and, utilizing a median-filtering-like process to adjust the previously inferred

[4] PLDA scoring computes the loglikelihood ratio between two embeddings

label (x_{t-1}). This process is performed with a window of the last three segments $W_t = (x_{t-2}, x_{t-1}, x_t)$, modifying the in-between speaker label if the surrounding labels are equal to each other, producing three contiguous segments with the same label.

3 Experiments

This section describes our experimental setup and results. We decided on a 1.0 s width and 0.5 s step sliding window at the speech segmentation step, discarding segments shorter than 0.5 s to ensure sufficient speaker information. Both i- and x-vectors were extracted using the Kaldi's CALLHOME diarization recipes[5] [18]. For CALLHOME x-vector experiments, a publicly available [20,21] model and PLDA backend were used.

3.1 Evaluation Metrics

The system performance was evaluated in terms of Equal Error Rate (EER) and minimum Detection Cost Function (minDCF) [25], as the key component of our tracking framework follows a speaker verification approach. Besides, we report Diarization Error Rate (DER) [5] since our framework shares characteristics with the standard diarization system.

3.2 Datasets

We tested our system on two standard public datasets: (1) CALLHOME, it contains 500 utterances distributed across six languages: Arabic, English, German, Japanese, Mandarin, and Spanish. Each utterance contains up to 7 speakers (2) DIHARD II single-channel development and evaluation subsets (LDC2019E31, LDC2019E32), focused on "hard" speaker diarization, contains 5-10 minute English utterances selected from 11 conversational domains, each including approximately 2 hours of audio. Since our approach is supervised, we performed a 2-fold cross-validation on each dataset using standard partitions: callhome1 and callhome2 from Kaldi's CALLHOME diarization recipe [18], and DIHARD II single channel's development and evaluation subsets. Then, the partitions' results are combined to report the averaged DER, EER and minDCF of each dataset.

To evaluate our proposed method in more difficult conditions, we increased the variability of the datasets in two steps. First, we increased the number of non-target speakers by adding to each recording speakers models from all the other recordings as new segments features. Such models were extracted with the same *model time* as the target's speakers. This set is used as the *speaker verification* conditions with its 0.17% target probability.

[5] https://github.com/kaldi-asr/kaldi/tree/master/egs/callhome diarization/v1 and /v2

The second modification to the datasets aims to give us a better hint of the system's performance in a real-life scenario, by increasing the number of overlapping speech instances, as both datasets have a low percentage of speaker overlap (CALLHOME ∼16%, DIHARD II single channel ∼9%). To increase the overlapping examples, we use the ground-truth labels to extract the non-overlapping audio segments of each speaker. Then, those segments are merged into a set of single-speaker utterances for each recording. After that, the single-speaker utterances are pairwise overlapped to create a new set of two-speaker-overlapping utterances. Finally, the new overlapping utterances are cut into segments (following Algorithm 1) and inserted into their original recordings at random locations with a uniform distribution.

Algorithm 1: Get the lengths to cut from an utterance

Result: A list of lengths to cut from an utterance
T is the length of an utterance;
L is an empty list;
while $T > 1.5$ **do**
    $l \leftarrow \sqrt{T}$;
    $T \leftarrow T - l$;
    append l to L;
end
append l to L;

The resulting datataset is used as the *speaker overlap* condition. It contains an additional ∼18% of speaker overlap in CALLHOME, and ∼30% in DIHARD II single channel. It is worth mentioning that the *speaker verification* condition is a subset of the *speaker overlap* one, so the target probability increases to 0.35% with the additional target examples.

3.3 Baseline

We compared the performance of our proposed system with a conventional offline diarization method: PLDA scoring with AHC, following the Kaldi's CALLHOME diarization recipe [18] with oracle number of speakers. The i- and x-vector PLDA backends were trained for each cross-validation fold with the recipe and used along all experiments.

Our primary baseline method follows the same procedure as our proposed system, but replaces the DNN-based identification module with a PLDA. The PLDA backends are the same as the ones used in the offline diarization baseline. We report the averaged results across the dataset partitions.

3.4 Results

The first set of experiments follows optimal conditions for speaker tracking: the input audio signal contains only speech from two tracked-speakers, and there is

Table 2. DER (%), EER (%) and minDCF (52% target probability) on two datasets given the optimal conditions.

Model time	PLDA			DNN		
	DER	**EER**	**minDCF**	**DER**	**EER**	**minDCF**
CALLHOME i-vector (Offline DER: 16.95)						
3.0 s	7.11	19.03	0.39	5.86	4.33	0.08
5.5 s	5.47	16.09	0.33	4.99	3.32	0.06
10.5 s	4.42	14.32	0.29	**4.30**	**2.84**	**0.05**
x-vector (Offline DER: 15.84)						
3.0 s	9.86	17.63	0.36	11.46	11.45	0.23
5.5 s	7.21	14.18	0.29	8.61	8.10	0.16
10.5 s	5.53	11.66	0.24	5.74	4.83	0.10
DIHARD II i-vector (Offline DER: 21.53)						
3.0 s	18.96	36.62	0.75	18.22	21.95	0.45
5.5 s	16.11	34.73	0.72	13.80	14.80	0.30
10.5 s	13.23	33.70	0.69	**11.36**	**12.45**	**0.26**
x-vector (Offline DER: 21.36)						
3.0 s	15.80	28.03	0.58	20.20	27.25	0.56
5.5 s	11.95	24.86	0.51	18.20	25.75	0.53
10.5 s	10.17	23.66	0.49	12.25	15.75	0.32

no overlapping speech. To have 2-speaker recordings, we applied a mask at the instances where a third speaker appeared in each recording.

We took this decision based on the fact that if we filtered out entire recordings with more than two speakers, we would have lost a large percentage of each dataset (60% CALLHOME and 47% DIHARD II single channel).

Table 2 show the results. All offline diarization results follow the same trend: x-vectors perform better than i-vectors, with the PLDA-based tracking having a clear advantage over it's offline counterpart. The reason behind this behavior is that the tracking pipeline receives the speakers' models beforehand.

An interesting phenomenon is that the PLDA-based tracking in CALLHOME shows better DER performance with i-vectors rather than x-vectors (also happens in Table 3). We believe that this is related to the generation of speakers models with embeddings trained with less data (as it does not happen in DIHARD II, whose x-vector extractor was trained with VoxCeleb data).

In most cases, the DNN-based tracking outperforms the PLDA baseline in the verification metrics (EER, minDCF). It is reasonable for several reasons: (1) The network's training promoted a binary-like similarity score. (2) Due to the speaker models permutations performed in training, the network had to perform more rejections. (3) The similarity score for each speaker is computed with all speakers' models available as contextual information.

For DER, the PLDA system has a clear advantage. Still, the DNN pipeline keeps close results despite its relatively simple architecture; we expect to overcome this by moving to a recurrent neural network (RNN).

The most interesting phenomenon in Tables 2 and 3 is that in all DNN results, the x-vectors have a clear disadvantage against i-vectors in all the provided metrics. We reviewed and discarded possible procedural and architectural mistakes. The same behavior was found in [12] with a similar DNN architecture. We agree that a possible reason for this behavior is the need for a complex DNN architecture to score an embedding derived from a much more complex architecture.

Table 3. DER (%), EER (%) and minDCF (17% target probability) given the speaker verification conditions.

Model time	PLDA			DNN		
	DER	**EER**	**minDCF**	**DER**	**EER**	**minDCF**
CALLHOME i-vector						
5.5 s	5.47	22.27	0.80	4.56	7.75	0.28
10.5 s	4.42	22.22	0.83	**4.43**	**5.89**	**0.21**
x-vector						
5.5 s	7.20	11.09	0.41	8.13	8.74	0.30
10.5 s	5.53	9.50	0.37	6.24	4.40	0.19
DIHARD II i-vector						
5.5 s	16.11	32.64	0.99	15.42	17.96	0.62
10.5 s	13.23	32.89	0.99	**11.44**	**14.67**	**0.54**
x-vector						
5.5 s	11.85	15.59	0.70	16.84	15.89	0.66
10.5 s	10.25	15.17	0.68	13.77	15.06	0.73

Finally, we evaluate our proposed system considering overlapped speech, as described in Section 3.2. In this set of experiments, the number of tracked speakers is fixed to 2, with the input audio signal containing non-overlapping and overlapping speech from them in addition to non-target speakers.

In order to select a segment as an overlap of the tracked speakers, it was necessary to train a DNN model able to work with three speaker models simultaneously ($S = 3$), the first two models representing each of the two speakers, and the third one, their overlap; as shown in Fig. 4. During test time, an embedding of the tracked speaker's overlapping speech was used as the third model, so when the network selected such embedding, we knew it was overlapping speech from the tracked speakers.

In 4, we report a loss in DER performance as overlapping speech dramatically increases the complexity of tracking. However, even with an additional model to score, we keep competitive performance in both EER and minDCF since DNN keeps its binary-like scoring while selecting overlapping speech.

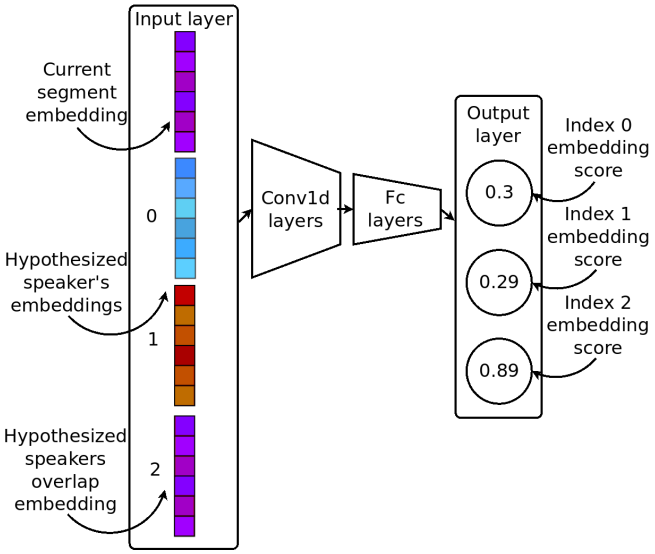

Fig. 4. Network input and output layers extended for overlap detection.

Table 4. DER (%), EER (%) and minDCF (35% target probability) for **i-vector**, given the speaker overlap conditions.

Model time	CALLHOME			DIHARD II		
	DER	**EER**	**minDCF**	**DER**	**EER**	**minDCF**
3.0 s	20.82	13.20	0.35	37.17	29.46	0.73
5.5 s	15.78	9.72	0.26	31.99	24.78	0.64
10.5 s	**12.52**	**7.50**	**0.20**	**28.78**	**21.32**	**0.55**

4 Conclusions

In this paper, we propose a novel embedding-based speaker-tracking DNN model focused on online tracking. We demonstrated our approach's efficiency through several experiments on two standard public datasets: CALLHOME and DIHARD II single channel. Results show better performance than the PLDA baseline in EER and minDCF in different experimental conditions.

For future research, we would like to extend our current DNN model to an **online** diarization and tracking system, where a recurrent neural network (RNN) will be responsible for selecting and updating the speaker models without having to resort to external sources. We expect such a system to provide not only the diarization results but also the set of speaker models that it will generate during an adaptive diarization process.

Acknowledgments. We would like to thank Diego Nigel Joaquin Campos Sobrino and Mario Alejandro Campos Soberanis for their helpful discussions.

References

1. Bonastre, J.F., Delacourt, P., Fredouille, C., Merlin, T., Wellekens, C.: A speaker tracking system based on speaker turn detection for NIST evaluation. Acoustics, Speech, and Signal Processing, IEEE International Conference on 2, II1177–II1180 (06 2000)
2. Dabbabi, K., Salah, H., Adnen, C.: Hybridization DE with k-means for speaker clustering in speaker diarization of broadcasts news. International Journal of Speech Technology 22 (09 2019)
3. Dehak, N., Kenny, P.J., Dehak, R., Dumouchel, P., Ouellet, P.: Front-end factor analysis for speaker verification. IEEE Transactions on Audio, Speech, and Language Processing 19(4), 788–798 (May 2011)
4. Diez, M., Landini, F., Burget, L., Rohdin, J., Silnova, A., Žmolíková, K., Novotny, O., Veselý, K., Glembek, O., Plchot, O., Mošner, L., Matejka, P.: BUT system for DIHARD speech diarization challenge 2018. pp. 2798–2802 (09 2018)
5. Fiscus, J., Ajot, J., Michel, M., Garofolo, J.: The rich transcription 2006 spring meeting recognition evaluation. pp. 309–322 (01 2006)
6. Garcia, P., Villalba, J., Bredin, H., Du, J., Castan, D., Cristia, A., Bullock, L., Guo, L., Okabe, K., Nidadavolu, P.S., Kataria, S., Chen, S., Galmant, L., Lavechin, M., Sun, L., Gill, M.P., Ben-Yair, B., Abdoli, S., Wang, X., Bouaziz, W., Titeux, H., Dupoux, E., Lee, K.A., Dehak, N.: Speaker detection in the wild: Lessons learned from JSALT 2019 (2019)
7. Ghahabi, O., Fischer, V.: Speaker-Corrupted Embeddings for Online Speaker Diarization. In: Proc. Interspeech 2019. pp. 386–390 (2019)
8. Karim, D., Adnen, C., Salah, H.: A system for speaker detection and tracking in audio broadcast news. In: 2017 International Conference on Engineering MIS (ICEMIS). pp. 1–5 (2017)
9. Khosravani, A., Homayounpour, M.M.: A PLDA approach for language and text independent speaker recognition. Computer Speech & Language 45, 457 – 474 (2017), http://www.sciencedirect.com/science/article/pii/S0885230816302972
10. Landini, F., Wang, S., Diez, M., Burget, L., Matějka, P., Žmolíková, K., Mošner, L., Plchot, O., Novotný, O., Zeinali, H., Rohdin, J.: BUT system description for DIHARD speech diarization challenge 2019 (2019)
11. Lin, Q., Yin, R., Li, M., Bredin, H., Barras, C.: LSTM based similarity measurement with spectral clustering for speaker diarization. Interspeech 2019 (Sep 2019)
12. Medennikov, I., Korenevsky, M., Prisyach, T., Khokhlov, Y., Korenevskaya, M., Sorokin, I., Timofeeva, T., Mitrofanov, A., Andrusenko, A., Podluzhny, I., Laptev, A., Romanenko, A.: Target-speaker voice activity detection: a novel approach for multi-speaker diarization in a dinner party scenario (2020)
13. Novoselov, S., Gusev, A., Ivanov, A., Pekhovsky, T., Shulipa, A., Avdeeva, A., Gorlanov, A., Kozlov, A.: Speaker diarization with deep speaker embeddings for DIHARD challenge II. In: INTERSPEECH (2019)
14. Pal, M., Kumar, M., Peri, R., Narayanan, S.: A study of semi-supervised speaker diarization system using GAN mixture model (2019)
15. Park, T.J., Han, K.J., Kumar, M., Narayanan, S.: Auto-tuning spectral clustering for speaker diarization using normalized maximum eigengap. IEEE Signal Processing Letters 27, 381–385 (2020)

16. Ryant, N., Church, K., Cieri, C., Cristia, A., Du, J., Ganapathy, S., Liberman, M.: The second DIHARD diarization challenge: Dataset, task, and baselines (2019)
17. Sell, G., Garcia-Romero, D.: Speaker diarization with PLDA i-vector scoring and unsupervised calibration. 2014 IEEE Workshop on Spoken Language Technology, SLT 2014 - Proceedings pp. 413–417 (04 2015)
18. Sell, G., Snyder, D., McCree, A., Garcia-Romero, D., Villalba, J., Maciejewski, M., Manohar, V., Dehak, N., Povey, D., Watanabe, S., Khudanpur, S.: Diarization is hard: Some experiences and lessons learned for the JHU team in the inaugural DIHARD challenge. pp. 2808–2812 (09 2018)
19. Snyder, D., Garcia-Romero, D., Sell, G., McCree, A., Povey, D., Khudanpur, S.: Speaker recognition for multi-speaker conversations using x-vectors. In: ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 5796–5800 (May 2019)
20. Snyder, D., Garcia-Romero, D., Sell, G., Povey, D., Khudanpur, S.: X-vectors: Robust DNN embeddings for speaker recognition. In: 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 5329–5333 (April 2018)
21. Snyder, D., Garcia-Romero, D., Sell, G., Povey, D., Khudanpur, S.: X-vectors: Robust dnn embeddings for speaker recognition. In: 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE (2018)
22. Snyder, D., Garcia-Romero, D., McCree, A., Sell, G., Povey, D., Khudanpur, S.: Spoken language recognition using x-vectors. pp. 105–111 (06 2018)
23. Snyder, D., Ghahremani, P., Povey, D., Garcia-Romero, D., Carmiel, Y., Khudanpur, S.: Deep neural network-based speaker embeddings for end-to-end speaker verification. pp. 165–170 (12 2016)
24. Sonmez, K., Heck, L., Weintraub, M.: Speaker tracking and detection with multiple speakers (September 1999), https://www.microsoft.com/en-us/research/publication/speaker-tracking-and-detection-with-multiple-speakers/
25. Van Leeuwen, D., Brummer, N.: An introduction to application-independent evaluation of speaker recognition systems. vol. 4343, pp. 330–353 (01 2007)
26. Variani, E., Lei, X., McDermott, E., Moreno, I., Gonzalez-Dominguez, J.: Deep neural networks for small footprint text-dependent speaker verification. pp. 4052–4056 (05 2014)
27. Wang, Q., Downey, C., Wan, L., Mansfield, P.A., Moreno, I.L.: Speaker diarization with LSTM (2017)
28. Xu, L., Das, R.K., Yılmaz, E., Yang, J., Li, H.: Generative x-vectors for text-independent speaker verification (2018)
29. Zajíc, Z., Kunešová, M., Hruz, M., Vanek, J.: UWB-NTIS speaker diarization system for the DIHARD II 2019 challenge (05 2019)
30. Zhang, A., Wang, Q., Zhu, Z., Paisley, J., Wang, C.: Fully supervised speaker diarization (2018)
31. Zhao, C., Li, L., Wang, D., Pu, A.: Local training for PLDA in speaker verification. In: 2016 Conference of The Oriental Chapter of International Committee for Coordination and Standardization of Speech Databases and Assessment Techniques (O-COCOSDA). pp. 156–160 (Oct 2016)