SPEAKER DIARIZATION WITH LSTM

Quan Wang[1]   Carlton Downey[2]   Li Wan[1]   Philip Andrew Mansfield[1]   Ignacio Lopez Moreno[1]

[1]Google Inc., USA   [2]Carnegie Mellon University, USA

[1]{ quanw, liwan, memes, elnota } @google.com   [2]cmdowney@cs.cmu.edu

ABSTRACT

For many years, i-vector based audio embedding techniques were the dominant approach for speaker verification and speaker diarization applications. However, mirroring the rise of deep learning in various domains, neural network based audio embeddings, also known as *d-vectors*, have consistently demonstrated superior speaker verification performance. In this paper, we build on the success of d-vector based speaker verification systems to develop a new d-vector based approach to speaker diarization. Specifically, we combine LSTM-based d-vector audio embeddings with recent work in non-parametric clustering to obtain a state-of-the-art speaker diarization system. Our system is evaluated on three standard public datasets, suggesting that d-vector based diarization systems offer significant advantages over traditional i-vector based systems. We achieved a 12.0% diarization error rate on NIST SRE 2000 CALLHOME, while our model is trained with out-of-domain data from voice search logs.

Index Terms— Speaker diarization, deep learning, audio embedding, LSTM, spectral clustering

1. INTRODUCTION

Speaker diarization is the process of partitioning an input audio stream into homogeneous segments according to the speaker identity. It answers the question "*who spoke when*" in a multi-speaker environment. It has a wide variety of applications including multimedia information retrieval, speaker turn analysis, and audio processing. In particular, the speaker boundaries produced by diarization systems have the potential to significantly improve acoustic speech recognition (ASR) accuracy.

A typical speaker diarization system usually consists of four components: (1) Speech segmentation, where the input audio is segmented into short sections that are assumed to have a single speaker, and the non-speech sections are filtered out; (2) Audio embedding extraction, where specific features such as MFCCs [1], speaker factors [2], or i-vectors [3, 4, 5] are extracted from the segmented sections; (3) Clustering, where the number of speakers is determined, and the extracted audio embeddings are clustered into these speakers; and optionally (4) Resegmentation [6], where the clustering results are further refined to produce the final diarization results.

In recent years, neural network based audio embeddings (d-vectors) have seen wide-spread use in speaker verification applications [7, 8, 9, 10, 11], often significantly outperforming previously state-of-the-art techniques based on i-vectors. However, most of these applications belong to text-dependent speaker verification, where the speaker embeddings are extracted from specific detected

---

More information of this work can be found at: https://google.github.io/speaker-id/publications/LstmDiarization

keywords [12, 13]. In contrast, speaker diarization requires text-independent embeddings which work on arbitrary speech.

In this paper, we explore a text-independent d-vector based approach to speaker diarization. We leverage the work of [11] to train an LSTM-based text-independent speaker verification model, then combine this model with recent work in non-parametric spectral clustering algorithm to obtain a state-of-the-art speaker diarization system.

While several authors have had explored using neural network embeddings for diarization tasks, their work has largely focused on using feed-forward DNNs to directly perform diarization. For example, [14] uses DNN embeddings trained on PLDA-inspired loss. In contrast, our work uses RNNs (specifically LSTMs [15]), which better capture the sequential nature of audio signals, and our generalized end-to-end training architecture directly simulates the enroll-verify run-time logic.

There have been several attempts to apply spectral clustering [16] to the speaker diarization problem [17, 3]. However, to the authors' knowledge, our work is the first to combine LSTM-based d-vector embeddings with spectral clustering. Furthermore, as part of our spectral clustering algorithm, we present a novel sequence of affinity matrix refinement steps which act to de-noise the affinity matrix, and are crucial to the success of our system.

The remainder of this paper is organized as follows: In Sec. 2, we describe how the LSTM-based text-independent speaker verification model trained with the framework in [11] can be adapted to featurize raw audio data and prepare it for clustering. In Sec. 3, we describe four different clustering algorithms and discuss the pros and cons of each in the context of speaker diarization, culminating with a modified spectral clustering algorithm. Experimental results and discussions are presented in Sec. 4, and conclusions are in Sec. 5.

2. DIARIZATION WITH D-VECTORS

Wan *et al.* recently introduced an LSTM-based [15] speaker embedding network for both text-dependent and text-independent speaker verification [11]. Their model is trained on fixed-length segments extracted from a large corpus of arbitrary speech. They showed that the d-vector embeddings produced by such networks usually significantly outperform i-vectors in an enrollment-verification 2-stage application. We now describe how this model can be modified for purposes of speaker diarization.

The flowchart of our diarization system is provided in Fig. 1. In this system, audio signals are first transformed into frames of width 25ms and step 10ms, and log-mel-filterbank energies of dimension 40 are extracted from each frame as the network input. We build sliding windows of a fixed length on these frames, and run the LSTM network on each window. The last-frame output of the LSTM is then used as the d-vector representation of this sliding window.
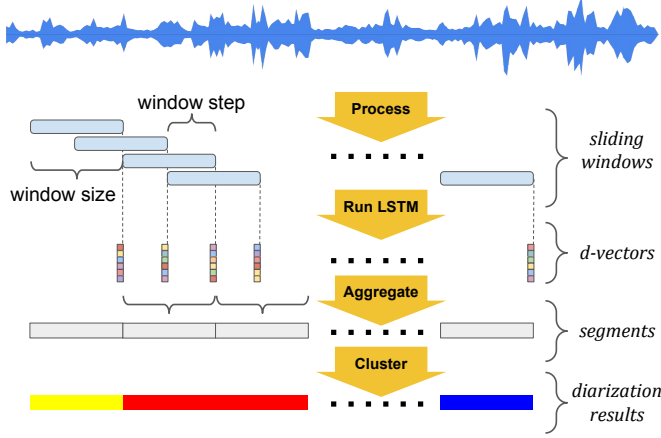
Fig. 1: A flowchart of our d-vector based diarization system.

We use a Voice Activity Detector (VAD) to determine speech segments from the audio, which are further divided into smaller *non-overlapping* segments using a maximal segment-length limit (*e.g.* 400ms in our experiments), which determines the temporal resolution of the diarization results. For each segment, the corresponding d-vectors are first L2 normalized, then averaged to form an embedding of the segment.

The above process serves to reduce arbitrary length audio input into a sequence of fixed-length embeddings. We can now apply a clustering algorithm to these embeddings in order to determine the number of unique speakers, and assign each part of the audio to a specific speaker.

3. CLUSTERING

In this section, we introduce the four clustering algorithms that we integrated into our diarization system. We place particular focus on the spectral offline clustering algorithm, which significantly outperformed the alternative approaches across experiments.

We note that clustering algorithms can be separated into two categories according to the run-time latency:

- **Online clustering**: A speaker label is immediately emitted once a segment is available, without seeing future segments.
- **Offline clustering**: Speaker labels are produced after the embeddings of all segments are available.

Offline clustering algorithms typically outperform Online clustering algorithms due to the additional contextual information available in the offline setting. Furthermore, a final resegmentation step can only be applied in the offline setting. Nonetheless, the choice between online and offline depends primarily on the nature of the application — where the system is intended to be deployed. For example, latency-sensitive applications such as live video analysis typically restrict the system to online clustering algorithms.

3.1. Naive online clustering

This is a prototypical online clustering algorithm. We apply a threshold on the similarities between embeddings of segments. To be consistent with the generalized end-to-end training architecture [11], *cosine similarity* is used as our similarity metric.

In this clustering algorithm, each cluster is represented by the centroid of all its corresponding embeddings. When a new segment embedding is available, we compute its similarities to centroids of all existing clusters. If they are all smaller than the threshold, then create a new cluster containing only this embedding; otherwise, add this embedding to the most similar cluster and update the centroid.

3.2. Links online clustering

Links is an online clustering method we developed to improve upon the naive approach. It estimates cluster probability distributions and models their substructure based on the embedding vectors received so far. The technical details are described in a separate paper [18].

3.3. K-Means offline clustering

Like in many diarization systems [19, 3, 20], we integrated the K-Means clustering algorithm with our system. Specifically, we use K-Means++ for initialization [21]. To determine the number of speakers $\widetilde{k}$, we use the "elbow" of the derivatives of conditional Mean Squared Cosine Distances[1] (MSCD) between each embedding to its cluster centroid:

$$\widetilde{k} = \arg\max_{k \geq 1} \text{MSCD}'(k). \tag{1}$$

3.4. Spectral offline clustering

Our spectral clustering algorithm consists of the following steps:

1. Construct the affinity matrix A, where A_{ij} is the cosine similarity between ith and jth segment embedding when $i \neq j$, and the diaginal elements are set to the maximal value in each row: $A_{ii} = \max_{j \neq i} A_{ij}$.

2. Apply the following sequence of refinement operations on the affinity matrix A:

   (a) Gaussian Blur with standard deviation σ;
   (b) Row-wise Thresholding: For each row, set elements smaller than this row's p-percentile to 0; [2]
   (c) Symmetrization: $Y_{ij} = \max(X_{ij}, X_{ji})$;
   (d) Diffusion: $Y = XX^T$;
   (e) Row-wise Max Normalization: $Y_{ij} = X_{ij} / \max_k X_{ik}$.

These refinements act to both smooth and denoise the data in the similarity space as shown in Fig. 2, and are crucial to the success of the algorithm. The refinements are based on the temporal locality of speech data — contiguous speech segments should have similar embeddings, and hence similar values in the affinity matrix.

We now provide the intuition behind each of these operations: The Gaussian blur acts to smooth the data, and reduce the effect of outliers. Row-wise thresholding serves to zero-out affinities between embeddings belonging to two different speakers. Symmetrization restores matrix symmetry which is crucial to the spectral clustering algorithm. The diffusion steps draws inspiration from the Diffusion Maps algorithm [22], and serves to sharpen the image resulting in clear boundaries between sections of the affinity matrix belonging to distinct speakers. Finally, the row-wise max normalization serves to rescale the spectrum of the matrix to ensure undesirable scale effects do not occur during the subsequent spectral clustering step.

---

[1] We define cosine distance as $d(x, y) = (1 - \cos(x, y))/2$.

[2] In practice, it's better to use soft thresholding: scale these elements by a small multiplier such as 0.01.
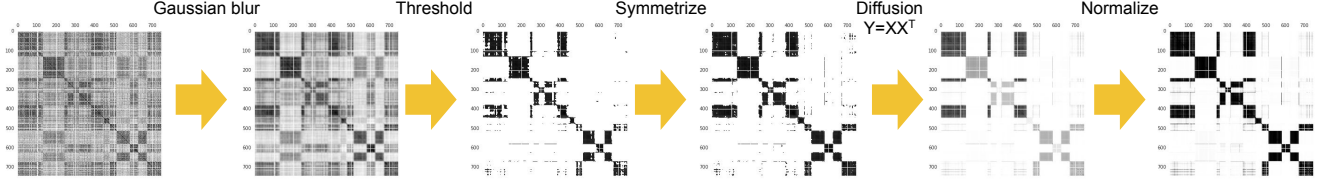
Fig. 2: Refinement operations on the affinity matrix.

3. After all refinement operations have been applied, perform eigen-decomposition on the refined affinity matrix. Let the n eigen-values be: $\lambda_1 > \lambda_2 > \cdots > \lambda_n$. We use the maximal eigen-gap to determine the number of clusters $\widetilde{k}$:

$$\widetilde{k} = \arg \max_{1 \leq k \leq n} \frac{\lambda_k}{\lambda_{k+1}}. \tag{2}$$

4. Let the eigen-vectors corresponding to the largest $\widetilde{k}$ eigen-values be $v_1, v_2, \cdots, v_{\widetilde{k}}$. We replace the ith segment embedding by the corresponding dimension in these eigen-vectors: $e_i = [v_{1i}, v_{2i}, \cdots, v_{\widetilde{k}i}]$. Then we use the same K-Means algorithm in Sec. 3.3 to cluster these new embeddings, and produce speaker labels.

3.5. Discussion

Speech data analysis is an extremely challenging problem domain, and conventional clustering algorithms such as K-Means often perform poorly. This is due to a number of unfortunate properties inherent to speech data, which include:

(i) **Non-Gaussian Distributions**: Speech data are often Non-Gaussion. In this setting, the centroid of a cluster (central to K-Means clustering) is not a sufficient representation.

(ii) **Cluster Imbalance**: In speech data, it is often the case that one speaker will speak often, while other speakers will speak rarely. In this setting, K-Means may incorrectly split large clusters into several smaller clusters.

(iii) **Hierarchical Structure**: Speakers fall into various groups according to gender, age, accent, *etc.* This structure is problematic since the difference between a male and a female speaker is much larger than the difference between two female speakers. This makes it difficult for K-Means to distinguish between clusters corresponding to groups, and clusters corresponding to distinct speakers. In practice, this often causes K-Means to incorrectly cluster all embeddings corresponding to male speakers into one cluster, and all embeddings corresponding to female speakers into another.

The problems caused by these properties are not limited to K-Means clustering, but are endemic to most parametric clustering algorithms. Fortunately, these problems can be mitigated by employing a non-parametric connection-based clustering algorithm such as spectral clustering.

4. EXPERIMENTS

4.1. Models

We run experiments with all combinations of both i-vector and d-vector models, with the four clustering algorithms discussed in Sec.

3. Both models are trained on an anonymized collection of voice searches, which has around 36M utterances and 18K speakers.

The i-vector model is trained using 13 PLP coefficients with delta and delta-delta coefficients. The GMM-UBM includes 512 Gaussians, and the total variability matrix includes 100 eigenvectors. The final i-vectors are reduced to 50-dimensional using LDA.

The d-vector model is a 3-layer LSTM network with a final linear layer. Each LSTM layer has 768 nodes, with projection [23] of 256 nodes.

Our Voice Activity Detection (VAD) model is a very small GMM model using the same PLP features as i-vector. It only has two full covariance Gaussians: one for speech, and one for non-speech. We found this simple VAD generalizes better across domains (from queries to telephone) for diarization than CLDNN [24] VAD models.

4.2. Datasets

We report Diarization Error Rates (DER) on three standard public datasets: (1) CALLHOME American English [25] (LDC97S42 + LDC97T14); (2) 2003 NIST Rich Transcription (LDC2007S10), the English conversational telephone speech (CTS) part; and (3) 2000 NIST Speaker Recognition Evaluation (LDC2001S97), Disk-8.

The first two datasets are English only, and are relatively smaller. Thus we use these two datasets to compare different algorithms.

The third dataset is used by most diarization papers, and is usually directly referred to as "CALLHOME" in literature. It contains 500 utterances distributed across six languages: Arabic, English, German, Japanese, Mandarin, and Spanish.

4.3. Experiment setup

Our diarization evaluation system is based on the pyannote.metrics library [26].

The CALLHOME American English dataset has a default 20-vs-20 utterances division for Dev-vs-Eval. For NIST RT-03 CTS, we randomly divide the 72 utterances into 14-vs-58 Dev and Eval sets. For each diarization system, we tune the parameters such as Voice Activity Detector (VAD) threshold, LSTM window size/step (Fig. 1), and clustering parameters on the Dev set, and report the DER on the Eval set.

For NIST RT-03 CTS, we only report DERs based on those provided un-partitioned evaluation map (UEM) files. For the other two datasets, as is the standard convention in literature [2, 3, 4, 6, 14, 27], we tolerate errors less than 250ms in locating segment boundaries.

As is typical, for each audio file, multiple channels are merged into a single channel [3, 6, 20], and we do not process the parts that are before the first annotation or after the last annotation. Additionally, as is standard in literature, we exclude overlapped speech (multiple speakers speaking at the same time) from our evaluation. For offline clustering algorithms, we constrain the system to produce at least 2 speakers.

Table 1: DER (%) on two English-only datasets for different embeddings and clustering algorithms.

Embedding	Clustering	CALLHOME American English Eval				NIST RT-03 English CTS Eval			
		Confusion	FA	Miss	Total	Confusion	FA	Miss	Total
i-vector	Naive	26.41			32.36	35.35			42.63
	Links	25.40	2.40	3.55	31.36	33.56	4.66	2.62	40.48
	K-Means	22.86			28.81	24.38			31.66
	Spectral	14.59			20.54	13.84			21.12
d-vector	Naive	12.41			18.87	18.76			27.30
	Links	11.02	1.94	4.51	17.47	18.56	4.09	4.45	27.10
	K-Means	7.29			13.75	7.80			16.34
	Spectral	6.03			12.48	3.76			12.30

Table 2: DER (%) on NIST SRE 2000 CALLHOME. Since we didn't do resegmentation, we report others' work by listing both with & without Variational Bayesian (VB) resegmentation [6]. Note that unlike others' work, our model is trained with out-of-domain data (English voice search vs. multilingual telephone speech).

Method	Confusion	FA	Miss	Total
d-vector + spectral	12.0	2.2	4.6	18.8
Castaldo et al. [2]	13.7	—	—	—
Shum et al. [3]	14.5	—	—	—
Senoussaoui et al. [4]	12.1	—	—	—
Sell et al. [6] (+VB)	13.7 (11.5)	—	—	—
Romero et al. [14] (+VB)	12.8 (9.9)	—	—	—

Table 3: DER (%) on CALLHOME American English 2-speaker subset (CH-109).

Method	Confusion	FA	Miss	Total
d-vector + spectral	5.97	2.51	4.06	12.54
Zajíc et al. [27]	7.84	—	—	—

4.4. Results

Our experimental results are shown in Table 1, 2 and 3. We report the total DER together with its three components: False Alarm (FA), Miss, and Confusion. FA and Miss are mostly from Voice Activity Detection errors, and partly from the aggregation from frame-level i-vectors or window-level d-vectors to segments. The FA and Miss differences between i-vector and d-vector are due to their different window sizes/steps and aggregation logics.

In Table 1, we can see that d-vector based diarization systems significantly outperform i-vector based systems. For d-vector systems, the optimal sliding window size and step are 240ms and 120ms, respectively.

We also observe that as expected, offline diarization produces significantly better results than online diarization. Specifically, on-line diarization predicts the incorrect number of speakers much more frequently than offline diarization. This problem could potentially be mitigated by the addition of a "burn-in" stage before entering the online mode.

In Table 2, we compare our d-vector + spectral clustering system with others' work on the same dataset. Though our LSTM model is completely trained on out-of-domain and English-only data, we can still achieve state-of-the-art performance on this multilingual dataset. The performance could potentially be further improved by using in-domain training data and adding a final resegmentation step.

Additionally, in Table 3, we followed the same practice in [27] to evaluate our system on a subset of 109 utterances from CALL-HOME American English that have 2 speakers (called CH-109 in [20]). Number of speakers is fixed to 2 for this evaluation.

4.5. Discussion

Though we listed DER metrics from different papers in Table 2 and 3, we find that it is difficult to fully align these numbers, an unfor-

tunately common problem in the diarization community. This is due primarily to the large number of moving parts required for a functional diarization pipeline. For example, different teams use different Voice Activity Detection marks (not publicly available), different training datasets, and different Dev sets for parameter tuning.

The evaluation protocols and software also differ from paper to paper. Most teams exclude FA and Miss from their evaluations, and directly refer to Confusion as their DER. However, we observed that a poor VAD with high Miss usually filters out the difficult parts in the speech, and makes the clustering problem much easier. Some papers like [20] use the non-standard Speaker Clustering Errors in frame percentage as their metric, and also exclude FA and Miss from this error. Additionally, it's unclear how overlapped speech is handled in some papers.

In our experiments, we do our best to ensure the comparisons are as fair as possible, and avoid tuning parameters on Eval sets.

5. CONCLUSIONS

In this paper, we built on the success of d-vector based speaker verification systems to develop a new d-vector based approach to speaker diarization. Specifically, we combined LSTM-based d-vector audio embeddings with recent work in non-parametric clustering to obtain a state-of-the-art speaker diarization system. We conducted experiments on four clustering algorithms combined with both i-vectors and d-vectors, and reported the performance on three standard public datasets: CALLHOME American English, NIST RT-03 English CTS, and NIST SRE 2000. In general, we observed that d-vector based systems achieve significantly lower DER than i-vector based systems.

6. ACKNOWLEDGEMENTS

7. REFERENCES

[1] Patrick Kenny, Douglas Reynolds, and Fabio Castaldo, "Diarization of telephone conversations using factor analysis," *IEEE Journal of Selected Topics in Signal Processing*, vol. 4, no. 6, pp. 1059–1070, 2010.

[2] Fabio Castaldo, Daniele Colibro, Emanuele Dalmasso, Pietro Laface, and Claudio Vair, "Stream-based speaker segmentation using speaker factors and eigenvoices," in *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*. IEEE, 2008, pp. 4133–4136.

[3] Stephen H Shum, Najim Dehak, Réda Dehak, and James R Glass, "Unsupervised methods for speaker diarization: An integrated and iterative approach," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 10, pp. 2015–2028, 2013.

[4] Mohammed Senoussaoui, Patrick Kenny, Themos Stafylakis, and Pierre 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.

[5] Gregory Sell and Daniel 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.

[6] Gregory Sell and Daniel Garcia-Romero, "Diarization resegmentation in the factor analysis subspace," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 4794–4798.

[7] Ehsan Variani, Xin Lei, Erik McDermott, Ignacio Lopez Moreno, and Javier Gonzalez-Dominguez, "Deep neural networks for small footprint text-dependent speaker verification," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 4052–4056.

[8] Yu-hsin Chen, Ignacio Lopez-Moreno, Tara N Sainath, Mirkó Visontai, Raziel Alvarez, and Carolina Parada, "Locally-connected and convolutional neural networks for small footprint speaker recognition," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[9] Georg Heigold, Ignacio Moreno, Samy Bengio, and Noam Shazeer, "End-to-end text-dependent speaker verification," in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 5115–5119.

[10] F A Rezaur Rahman Chowdhury, Quan Wang, Li Wan, and Ignacio Lopez Moreno, "Attention-based models for text-dependent speaker verification," *arXiv preprint arXiv:1710.10470*, 2017.

[11] Li Wan, Quan Wang, Alan Papir, and Ignacio Lopez Moreno, "Generalized end-to-end loss for speaker verification," *arXiv preprint arXiv:1710.10467*, 2017.

[12] Guoguo Chen, Carolina Parada, and Georg Heigold, "Small-footprint keyword spotting using deep neural networks," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 4087–4091.

[13] Rohit Prabhavalkar, Raziel Alvarez, Carolina Parada, Preetum Nakkiran, and Tara N Sainath, "Automatic gain control and multi-style training for robust small-footprint keyword spotting with deep neural networks," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 4704–4708.

[14] Daniel Garcia-Romero, David Snyder, Gregory Sell, Daniel Povey, and Alan 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.

[15] Sepp Hochreiter and Jürgen Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[16] Ulrike Von Luxburg, "A tutorial on spectral clustering," *Statistics and computing*, vol. 17, no. 4, pp. 395–416, 2007.

[17] Huazhong Ning, Ming Liu, Hao Tang, and Thomas S Huang, "A spectral clustering approach to speaker diarization.," in *INTERSPEECH*, 2006.

[18] Philip Andrew Mansfield, Quan Wang, Carlton Downey, Li Wan, and Ignacio Lopez Moreno, "Links: A high-dimensional online clustering method," *arXiv preprint arXiv:1801.10123*, 2018.

[19] Oshry Ben-Harush, Ortal Ben-Harush, Itshak Lapidot, and Hugo Guterman, "Initialization of iterative-based speaker diarization systems for telephone conversations," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 2, pp. 414–425, 2012.

[20] Dimitrios Dimitriadis and Petr Fousek, "Developing on-line speaker diarization system," in *INTERSPEECH*, 2017.

[21] David Arthur and Sergei Vassilvitskii, "k-means++: The advantages of careful seeding," in *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*. Society for Industrial and Applied Mathematics, 2007, pp. 1027–1035.

[22] Ronald R Coifman and Stéphane Lafon, "Diffusion maps," *Applied and computational harmonic analysis*, vol. 21, no. 1, pp. 5–30, 2006.

[23] Haşim Sak, Andrew Senior, and Françoise Beaufays, "Long short-term memory recurrent neural network architectures for large scale acoustic modeling," in *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.

[24] Rubén Zazo Candil, Tara N Sainath, Gabor Simko, and Carolina Parada, "Feature learning with raw-waveform cldnns for voice activity detection," 2016.

[25] A Canavan, D Graff, and G Zipperlen, "Callhome american english speech ldc97s42," *LDC Catalog. Philadelphia: Linguistic Data Consortium*, 1997.

[26] Hervé Bredin, "`pyannote.metrics`: a toolkit for reproducible evaluation, diagnostic, and error analysis of speaker diarization systems," *hypothesis*, vol. 100, no. 60, pp. 90, 2017.

[27] Zbyněk Zajíc, Marek Hrúz, and Luděk Müller, "Speaker diarization using convolutional neural network for statistics accumulation refinement," in *INTERSPEECH*, 2017.