Online Speaker Diarization Using Optimized SE-ResNet Architecture

Frantisek Kynych(✉), Jindrich Zdansky, Petr Cerva, and Lukas Mateju

Faculty of Mechatronics, Informatics and Interdisciplinary Studies,
Technical University of Liberec, Studentska 2, 461 17 Liberec, Czech Republic
frantisek.kynych@tul.cz

Abstract. A new approach to speaker diarization (SD) suitable for real-time processing of streamed data is presented in this work. It utilizes a modified residual network with squeeze-and-excitation blocks (SE-ResNet-34) for extraction of speaker embeddings. These speaker embeddings are calculated in an optimized way by using cached buffers and are subsequently used for voice activity detection (VAD) as well as for block-online k-means clustering with a look-ahead mechanism. All these processing steps are first evaluated separately on a development set compiled from recordings of Czech broadcast programs. The whole scheme is then compared to an offline reference approach on various speech databases that are publicly available and include data in various languages. On this data, our method yields results similar to the reference system while operating on a CPU with a low real-time factor (RTF) below 0.1 and a latency of around 5.5 s.

Keywords: Online speaker diarization · speaker embeddings · SE-ResNet · k-means clustering

1 Introduction

Speaker diarization (SD) is a process that answers the question "who spoke when" in a multi-speaker environment. Basically, two main possibilities exist for performing this task: in a) offline or b) streaming (online) mode. The input to the former (classic) scenario is usually formed by one speech recording. Its entire content can be processed without any strict limitations on computational demands, e.g., multiple passes through the data can be performed.

But today's world is accelerating; the data processing and information mining domains face a new challenge when their users ask for very quick results and analysis, ideally during the data flow. The increasing amount of data is organized into streams, which must be processed continuously.

Media monitoring is one of the typical applications where streamed data is processed. An example of such an application is our cloud platform for real-time transcription of TV and radio stations in several languages, including Czech, Slovak, Polish, and other predominantly Slavic languages.

K. Ekštein et al. (Eds.): TSD 2023, LNAI 14102, pp. 176–187, 2023.
https://doi.org/10.1007/978-3-031-40498-6_16

In this case, a diarization system allowing for real-time processing of the data streams must be employed. This system has to operate differently from its offline counterpart: it must be able to take in a sequence (stream) of frames on its input and provide a stream of speaker tags on its output. In consequence, there are additional limitations regarding namely the complexity of and computation demands on the used approach. Another important factor is latency: systems with a latency of around several seconds are considered to be online. In this case, there is an additional limitation on the context that can be processed in a given time step. Moreover, while offline SD can be improved by determining the number of speakers appearing in the data [2], this option is not available in most of our streamed scenarios (see Sect. 2).

In this work, we propose a new SD approach suitable for the above-mentioned real-time applications. Our approach processes the input data stream and produces a sequence of speaker embeddings on its output using SE-ResNet architecture optimized for online processing. These vectors are then filtered by a built-in voice activity detection module based on a single-layer binary classifier, and the remaining speech vectors are smoothed and clustered by the block-online k-means algorithm with a look-ahead mechanism.

At first, we evaluate and analyze the performance of individual phases of our method on a development set compiled from Czech TV/R recordings. Given all findings, this method is further evaluated on several publicly available datasets, including broadcast recordings in many languages.

2 Related Work

The early online SD approaches utilized hidden Markov models or Gaussian mixture models [11,34], and features such as the speaker factors [4]. More recently, the features used for online SD required a more robust speaker representation. Therefore, the i-vectors based on the total variability factor analysis began to be used [9,19,34].

These approaches were then surpassed by speaker embeddings produced by deep neural network architectures. These include d-vectors extracted mostly by long short-term memory recurrent neural networks [30,35] and x-vectors from the time-delay neural networks (TDNNs)[10].

The use of speaker embedding enables the option to perform diarization using various clustering algorithms. It is possible to use methods such as k-means [9,30], online naive clustering [30], or VBx algorithm with core samples selection [33]. Alternatively, a supervised model such as UIS-RNN [10,35] generating a sequence of speaker indices can be used instead of the conventional clustering. In addition to the aforementioned clustering-based diarization methods, recent work [31] has utilized a transformer transducer for detecting a change in speaker, extracting embeddings to represent speaker turns and clustering them using spectral clustering.

There has been a growing interest in end-to-end online diarization (EEND) approaches instead of the modular structure in recent years. Recent models are

based on an x-vector extractor with incremental clustering [6], encoder-decoder-attractor-EEND architecture with either a speaker-tracing buffer [32] or an incremental transformer encoder [13]. These techniques can handle overlapped speech and have overcome the limitation of having a variable number of speakers. The EEND approach is currently limited by the amount of data from the target domain needed for training, and its performance gets significantly lower with a larger number of speakers.

3 Proposed Approach

Our method utilizes the optimized SE-ResNet-34 [14] architecture for the extraction of speaker embeddings. These embeddings are then used for VAD as well as for clustering. These three steps are all described in the following subsections.

3.1 Speaker Embedding Extraction

We introduce two key optimizations to the SE-ResNet-34 topology (see also Table 1). Firstly, the SE-blocks in the model incorporate buffers consisting of the last two vectors from the previously processed data. These buffers are concatenated to the input at the beginning of the subsequent time step. Secondly, we apply the stride operation even in the first set of SE-blocks, exclusively affecting the feature dimension while keeping the time dimension unchanged. A combination of both of these optimizations allows us to calculate one speaker embedding for every feature vector from the input stream with an RTF factor lower by an order of magnitude (see also Sect. 3.4). These embeddings are produced per block of the input signal, and their values are the same as if they were calculated within the conventional offline scenario.

The number of the SE-blocks is the same as in the ResNet-34 architecture, and their utilization adds global context information by weighting the channels of feature maps. Convolution layers are conventionally followed by batch normalization and ReLU activation function. In contrast to the SE-ResNet-34, we do not utilize the attention mechanism because it does not yield any performance gain on our development set.

After the optimized SE-blocks, local pooling is used to compute the means and variations of the frames, with a context of $t \pm 20$ frames. These features are fed to a fully connected layer from which the speaker embeddings are extracted. The model is trained using the AM-Softmax loss [29] to distinguish between N speakers. As input features, a 256-point log magnitude spectrogram is computed from every frame of the input signal. These spectrograms are locally mean-normalized (LFMN) over a sliding window with the context of $t \pm 40$ frames. The length of each frame is 25 ms with a shift of 12.5 ms.

Table 1. Structure of the proposed optimized SE-ResNet extractor. T stands for the input size ($2 \times 93 + 1$ in our case).

Stage	Kernel size	Stride	Output Size
LFMN	–	–	$256 \times T \times 1$
Cached Conv	$3 \times 3 \times 32$	1	$256 \times T \times 32$
Cached Res1	$3 \times 3 \times 32$	(4, 1)	$64 \times T \times 32$
Cached Res2	$3 \times 3 \times 64$	(4, 1)	$16 \times T \times 64$
Cached Res3	$3 \times 3 \times 128$	(4, 1)	$4 \times T \times 128$
Cached Res4	$3 \times 3 \times 256$	(4, 1)	$1 \times T \times 256$
Pooling	–	–	$T \times 512$
Linear	–	–	$T \times 512$

3.2 Voice Activity Detection

The proposed approach incorporates a computationally undemanding mechanism for voice activity detection. This method utilizes a simple binary classifier with one fully connected layer. This network is trained using the binary cross-entropy loss function. The input to the classifier is formed by a single speaker embedding without any additional context, and the output is smoothed with the aid of moving average smoothing.

The key point here is that we utilize one additional speaker representing a non-speech class during training of the above-mentioned embedding extractor. The embeddings representing the non-speech class then form one cluster, and the corresponding segments of the input signal can be filtered out using a single-layer classifier. Experimental evaluation of the described VAD module is presented in Sect. 5.3.

3.3 Block-Online K-Means Clustering with Look-Ahead

We apply a block-online k-means algorithm to cluster speakers using the speaker embeddings extracted by the optimized SE-ResNet architecture. We employ cosine distance in the clustering process as the AM-Softmax (used within the training of the embedding extractor) computes speaker probabilities based on the same distance measure.

To avoid high sensitivity of the clustering, first, the embeddings are smoothed with the aid of moving average within the context of $t \pm 40$. After smoothing, conventional k-means clustering is performed on a part of the input stream. Two parameters determine the size of this part: block size and look-ahead size. The block size corresponds to the number of vectors to which the speaker tags will be assigned in a given step of the diarization process, while look-ahead size states how many additional future (non-causal) vectors are used within the clustering process to improve its accuracy. The size of the data used for clustering is thus

the block size plus look-ahead size. Note that each of the resulting clusters is represented by its centroid.

In the next step, we take into account only the resulting clusters whose numbers of associated vectors (embeddings) are higher than a defined threshold T_1. For each of these clusters, we compute its cosine distance from all of the existing centroids. If the distance to the closest existing centroid $c_{closest}$ is smaller than a threshold T_2 then the existing centroid is updated using linear interpolation with parameter α as $c_{closest} = (1 - \alpha)c_{closest} + \alpha c_{new}$. The remaining clusters with distances larger than T_2 represent new speakers.

The initial clusters are determined by a step size parameter. For example, if this value is set to 150, then every 150^{th} embedding in the input sequence forms an initial cluster. Finally, all vectors within the given block (determined by the block size) are assigned the appropriate speaker tags according to their affiliations with individual existing clusters.

3.4 Latency and Real-Time Factor

The latency of the proposed clustering is mainly given by the block size and by the non-causal look-ahead mechanism. For example, the values of these two parameters that were established during the development process correspond to a latency value of 4.4 s. The next source of latency is the non-causal part of the context used by the embedding extractor. Its size is $t \pm 93$ frames, which creates an extraction latency of 1.17 s. The total latency of the proposed diarization scheme is thus around 5.5 s.

At the same time, it operates with an RTF value of around 0.06 on a CPU (measured on Intel® Core™ i7 CPU 9700K CPU @ 3.60 GHz using one thread) while the original SE-ResNet-34 achieves RTF around 1.1 on NVIDIA® GeForce GTX 1080 Ti. The RTF is computed as the ratio of processing time to real-time duration.

4 Experimental Setup

4.1 Development Data

A dataset covering 12.7 h of broadcast data in the Czech language is used for development purposes. It consists of 51 files with recordings containing a minimum of 2 speakers and a maximum of 15 speakers (4.2 speakers on average). These recordings contain both clean speech segments and segments with music, background noise, jingles, and advertisements.

4.2 Evaluation Metrics

The equal error rate (EER) is employed for a comparison of different speaker embedding extractors in the speaker verification task. The diarization accuracy of our system on the development data is measured by word-level diarization

error rate (WDER). The motivation for using this metric stems from the fact that it is more important for our target application to assign the word to the correct speaker than to retrieve the exact time of the speaker change point. WDER represents the percentage of words with the correct speaker assigned.

Moreover, in Sect. 6, the standard diarization error rate (DER) is used for the comparison on other datasets that are publicly available. The DER consists of false alarm, missed speech, and speaker confusion and is computed using version 1.1.0 of the dscore[1] tool without any forgiveness collar. It also includes overlapped speech segments.

4.3 Reference System for Diarization

The diarization system is based on the Speechbrain (version 0.5.13) approach [7], which utilizes the ECAPA-TDNN [8] for embedding extraction, followed by spectral clustering. The embedding extractor uses 80-dimensional log Mel filterbank energies from the recording and mean normalizes them in the current segment. These features are extracted with a sliding window with a length of 1.5 s and a 0.5-second shift. After the embedding extraction, we use the unnormalized spectral clustering. The dataset used for training the ECAPA-TDNN model is the same as for our approach (see Sect. 5.1).

5 Experimental Evaluation

5.1 Speaker Embedding Extraction

In the first experiment, we compare the results on the speaker verification task of the original SE-ResNet-34 architecture, the proposed optimized SE-ResNet topology, and the ECAPA-TDNN reference system. All of these systems have been trained using the same data. This fact allows us to compare them directly.

The training data consists of VoxCeleb2 [5], "train-clean-360" subset of LibriSpeech [24], Czech microphone recordings, and part of CHiME-4 dataset [28] for the non-speech class. The LibriSpeech and Czech data have also been augmented with a combination of noise and reverberation, similar to that described in [21]. During training, the audio was randomly augmented with the MUSAN corpus [26] and with room impulse response simulations of small and medium rooms from [15]. A total number of 7,838 speakers have been used for training, where one additional class has represented noises.

The SE-ResNet model has been trained within 12 epochs using the AdamW optimizer with a learning rate of 0.003 and default torch parameters. We have employed the step learning rate decay with a 0.1 gamma value and lowered the learning rate every 5 epochs. The margin has been set to 0.3 and the scale factor to 15 in the AM-Softmax.

The datasets used for the evaluation represent the cleaned VoxCeleb1-E (extended), VoxCeleb1-H (hard) [23], TIMIT [12] and its augmented versions.

[1] https://github.com/nryant/dscore.

The applied augmentation strategies on TIMIT gradually increase the complexity of the speaker verification task. The original TIMIT version contains only noiseless signals. The Anechoic variant then includes anechoic and reverberated signals. These two augmentations are described in depth in [21]. The next Codecs version is described in [22], where the dataset is copied seven times, and different codecs are used for the augmentation of each copy. The last and most difficult Noisy version combines reverberation and noise for augmentation as proposed in [20].

The obtained results are compared in Table 2. The proposed online SE-ResNet architecture yields similar results as the original offline ResNet-34 topology and the reference ECAPA-TDNN system for the original TIMIT and its Anechoic version. At the same time, it has worse performance on both VoxCeleb datasets and TIMIT with more difficult augmentations, which is caused by its lower number of parameters.

Table 2. EER [%] for different architectures yielded in the speaker verification task on the VoxCeleb, TIMIT and its several augmented versions.

Datasets	SE-ResNet-34	proposed	ECAPA-TDNN
	offline	online	offline
VoxCeleb1-E	**1.61**	2.67	1.64
VoxCeleb1-H	3.14	4.34	**3.12**
orig. TIMIT	0.54	**0.16**	0.22
Anechoic	**0.19**	0.27	0.26
Codecs	0.58	1.53	**0.52**
Noisy	**1.48**	3.28	1.51

5.2 Block-Online Clustering

For the clustering, we have set α to 0.1 and T_1 to 149. The threshold T_2 for merging clusters has been 0.5. As mentioned in Sect. 3.3, the step size parameter for cluster initialization has been 150. All these parameters have been found on the development set in a series of experiments not presented in this paper.

Given these parameters, we have further investigated the effect of different block and look-ahead sizes as both of these parameters are important with regard to the latency. The block size has varied from 100 to 200 speaker embeddings and the look-ahead size from 150 to 250. We have also performed experiments with no look-ahead.

The obtained results (see Table 3) show that not using the look-ahead mechanism considerably worsens the performance of our system. The lowest WDER is achieved for the block size of 150 and the look-ahead size of 200. Both these values cause a latency of 4.4 s.

Table 3. WDER [%] on Czech broadcast recordings for different values of block size and look-ahead size.

Block size	Look-ahead size	WDER [%]
100	200	3.7
	250	4.1
150	0	12.3
	150	4.9
	200	**2.8**
	250	3.3
200	150	5.9
	200	3.7

5.3 Voice Activity Detection

The last experiment performed on the development set investigates the use of the VAD module with a binary classifier. For its training, 30 h of clean speech, 30 h of music, and 30 h of artificially mixed speech and music/noise recordings according to randomly chosen signal-to-noise ratio (SNR) have been used. All these recordings have also been concatenated in a random order to contain speech/non-speech transitions. Music recordings and the segments with SNR values smaller than 0 dB have been labeled as non-speech and the rest as speech.

The obtained results are presented in Table 4. Here, the VAD module without any smoothing slightly increases WDER from 2.8% to 2.9%. The reason is that the output decisions are too sensitive to noise in this case, and the module produces a lot of short speech segments. On the contrary, when VAD decisions are smoothed using the moving average filter with the context of 50 frames, the value of WDER is considerably decreased to 2.3%. Finally, it should also be noted that smoothing does not increase the latency of the whole diarization scheme. The reason is that it is not applied on the last 50 frames of the look-ahead data block during the clustering process.

Table 4. WDER [%] on Czech broadcast recordings with and without the VAD module.

Architecture	VAD	WDER [%]
SE-ResNet	none	2.8
	proposed	2.9
	proposed + MA	**2.3**

6 Results on other Datasets

The last section presents a comparison with the offline ECAPA-TDNN reference system. For this purpose, several broadcast datasets have been selected. The COST278 [27] database contains broadcast news in eleven European languages. The RTVE2018 [17] and RTVE2020 [18] databases contain recordings of various Spanish TV shows, including broadcast news, live magazines, quiz shows, or documentary series. Last, the RUNDKAST [1] is compiled from recordings of Norwegian broadcast news.

The results of the performed experiments recorded in Table 5 show that our optimized SE-ResNet system yields lower DER on the COST278, RUNDKAST, and RTVE2020 databases (e.g., 16.0% vs. 21.9% on RTVE2020 with VAD) and achieves slightly worse performance on the RTVE2018 dataset (i.e., 9.2% vs. 8.8% with applied VAD). These results show that our proposed architecture allows us to perform SD in streamed data with limited context while yielding performance comparable to the ECAPA-TDNN reference system.

Table 5. DER [%] results of the offline ECAPA-TDNN architecture and our proposed SE-ResNet online architecture on various datasets.

Dataset	VAD	ECAPA-TDNN	proposed
		offline	online
COST278	proposed	14.2	13.4
	ground-truth	12.6	**10.7**
RTVE2018	proposed	11.0	11.7
	ground-truth	**8.8**	9.2
RTVE2020	proposed	24.0	18.8
	ground-truth	21.9	**16.0**
RUNDKAST	proposed	13.4	13.2
	ground-truth	10.1	**9.7**

Finally, we have evaluated the proposed system also on datasets that are a bit far from our target domain but widely used in the community: the AMI meeting corpus [3] and DIHARD II [25] dataset. In the former case, the AMI full Mix-Headset evaluation protocol proposed in [16] is employed. The AMI evaluation uses the same clustering parameters as the previous experiments. For DIHARD II, the clustering context is smaller with block size set to 100, look-ahead to 100, and T_2 threshold to 0.35. These values have been found on the development set, resulting in a smaller latency of around 3.6 s.

The results in Table 6 show that our method achieves results comparable to other existing methods, but there is room for further improvement. This holds, namely in the processing of segments containing overlapping speech, which were the source of most of the errors and do not occur to such a large extent in our

target broadcast data. However, our diarization system has the advantage of requiring only one CPU core, while other systems require more computational resources, such as multiple CPU cores or GPUs. For example, the most powerful system [33] achieves an RTF of 0.1 using an NVIDIA® Geforce RTX 3090 GPU.

Table 6. DER [%] results on AMI and DIHARD II test sets.

Dataset	System				
	Proposed	[10]	[33]	[6]	[32]
AMI	21.2	–	**19.0**	27.5	–
DIHARD II	28.2	27.3	**23.1**	34.1	25.8

7 Conclusions

This work has focused on SD in streamed data. For this purpose, a new approach has been proposed. It consists of three consecutive phases. In the first one, speaker embeddings are extracted using SE-ResNet architecture, which is optimized by adding buffers and limited application of the stride. Then the VAD is applied, which utilizes the extracted embeddings and filters them using a single-layer binary classifier, whose output decisions are smoothed. The third (last) step makes use of block-online k-means clustering with a built-in look-ahead mechanism.

We compared our diarization scheme with a recent offline ECAPA-TDNN-based reference system on various broadcast datasets as well as with other online approaches on the out of domain but widely used AMI and DIHARD II datasets. All of the achieved results have demonstrated that the proposed method yields solid results. At the same time, it is capable of processing the streamed data just on a CPU with a low real-time factor below 0.1 and with a total latency of around 5.5 s.

Acknowledgements. The research leading to these results has received funding from the EEA / Norway Grants and the Technology Agency of the Czech Republic within the KAPPA Programme (project No. TO01000027) and from the Student Grant Competition of the Technical University of Liberec under project No. SGS-2022-3052.

References

1. Amdal, I., Strand, O.M., Almberg, J., Svendsen, T.: RUNDKAST: an annotated Norwegian broadcast news speech corpus. In: LREC (2008)
2. Aronowitz, H., Solewicz, Y.A., Toledo-Ronen, O.: Online two speaker diarization. In: Odyssey, pp. 122–129 (2012)

3. Carletta, J., Ashby, S., Bourban, S., Flynn, M., Guillemot, M., et al.: The AMI meeting corpus: a pre-announcement. In: MLMI, pp. 28–39 (2005)
4. Castaldo, F., Colibro, D., Dalmasso, E., Laface, P., Vair, C.: Stream-based speaker segmentation using speaker factors and eigenvoices. In: ICASSP, pp. 4133–4136 (2008)
5. Chung, J.S., Nagrani, A., Zisserman, A.: VoxCeleb2: deep speaker recognition. In: Interspeech, pp. 1086–1090 (2018)
6. Coria, J.M., Bredin, H., Ghannay, S., Rosset, S.: Overlap-aware low-latency online speaker diarization based on end-to-end local segmentation. In: ASRU, pp. 1139–1146 (2021)
7. Dawalatabad, N., Ravanelli, M., Grondin, F., Thienpondt, J., Desplanques, B., Na, H.: ECAPA-TDNN embeddings for speaker diarization. In: Interspeech, pp. 3560–3564 (2021)
8. Desplanques, B., Thienpondt, J., Demuynck, K.: ECAPA-TDNN: emphasized channel attention, propagation and aggregation in TDNN based speaker verification. In: Interspeech, pp. 3830–3834 (2020)
9. Dimitriadis, D., Fousek, P.: Developing on-line speaker diarization system. In: Interspeech, pp. 2739–2743 (2017)
10. Fini, E., Brutti, A.: Supervised online diarization with sample mean loss for multi-domain data. In: ICASSP, pp. 7134–7138 (2020)
11. Friedland, G., Janin, A., Imseng, D., Miro, X.A., Gottlieb, L.R., et al.: The ICSI RT-09 speaker diarization system. IEEE Trans. Speech Audio Process. **20**(2), 371–381 (2012)
12. Garofolo, J.S.: TIMIT acoustic-phonetic continuous speech corpus. Linguistic Data Consortium (1993)
13. Han, E., Lee, C., Stolcke, A.: BW-EDA-EEND: streaming END-TO-END neural speaker diarization for a variable number of speakers. In: ICASSP, pp. 7193–7197 (2021)
14. Heo, H.S., Lee, B., Huh, J., Chung, J.S.: Clova baseline system for the VoxCeleb speaker recognition challenge 2020. CoRR abs/2009.14153 (2020)
15. Ko, T., Peddinti, V., Povey, D., Seltzer, M.L., Khudanpur, S.: A study on data augmentation of reverberant speech for robust speech recognition. In: ICASSP, pp. 5220–5224 (2017)
16. Landini, F., Profant, J., Diez, M., Burget, L.: Bayesian HMM clustering of x-vector sequences (VBx) in speaker diarization: theory, implementation and analysis on standard tasks. Comput. Speech Lang. **71**, 101254 (2022)
17. Lleida, E., Ortega, A., Miguel, A., Bazan, V., Perez, C., et al.: RTVE2018 database description (2018)
18. Lleida, E., Ortega, A., Miguel, A., Bazan-Gil, V., Perez, C., et al.: RTVE2020 database description (2020)
19. Madikeri, S.R., Himawan, I., Motlicek, P., Ferras, M.: Integrating online i-vector extractor with information bottleneck based speaker diarization system. In: Interspeech, pp. 3105–3109 (2015)
20. Malek, J., Jansky, J., Kounovsky, T., Koldovsky, Z., Zdansky, J.: Blind extraction of moving audio source in a challenging environment supported by speaker identification via x-vectors. In: ICASSP, pp. 226–230 (2021)
21. Malek, J., Zdansky, J.: Voice-activity and overlapped speech detection using x-vectors. In: TSD, pp. 366–376 (2020)
22. Malek, J., Zdansky, J., Cerva, P.: Robust recognition of conversational telephone speech via multi-condition training and data augmentation. In: TSD, pp. 324–333 (2018)

23. Nagrani, A., Chung, J.S., Zisserman, A.: VoxCeleb: a large-scale speaker identification dataset. In: Interspeech, pp. 2616–2620 (2017)
24. Panayotov, V., Chen, G., Povey, D., Khudanpur, S.: LibriSpeech: an ASR corpus based on public domain audio books. In: ICASSP, pp. 5206–5210 (2015)
25. Ryant, N., Church, K., Cieri, C., Cristia, A., Du, J., et al.: The second DIHARD diarization challenge: dataset, task, and baselines. In: Interspeech, pp. 978–982 (2019)
26. Snyder, D., Chen, G., Povey, D.: MUSAN: a music, speech, and noise corpus. CoRR abs/1510.08484 (2015)
27. Vandecatseye, A., Martens, J., Neto, J.P., Meinedo, H., Garcia-Mateo, C., et al.: The COST278 pan-European broadcast news database. In: LREC (2004)
28. Vincent, E., Watanabe, S., Nugraha, A.A., Barker, J., Marxer, R.: An analysis of environment, microphone and data simulation mismatches in robust speech recognition. Comput. Speech Lang. **46**, 535–557 (2017)
29. Wang, F., Cheng, J., Liu, W., Liu, H.: Additive margin softmax for face verification. IEEE Signal Process. Lett. **25**(7), 926–930 (2018)
30. Wang, Q., Downey, C., Wan, L., Mansfield, P.A., Lopez-Moreno, I.: Speaker diarization with LSTM. In: ICASSP, pp. 5239–5243 (2018)
31. Xia, W., et al.: Turn-to-diarize: online speaker diarization constrained by transformer transducer speaker turn detection. In: ICASSP, pp. 8077–8081 (2022)
32. Xue, Y., Horiguchi, S., Fujita, Y., Takashima, Y., Watanabe, S., et al.: Online streaming end-to-end neural diarization handling overlapping speech and flexible numbers of speakers. In: Interspeech, pp. 3116–3120 (2021)
33. Yue, Y., Du, J., He, M., Yeung, Y.T., Wang, R.: Online speaker diarization with core samples selection. In: Interspeech, pp. 1466–1470 (2022)
34. Zelenak, M., Schulz, H., Hernando, J.: Speaker diarization of broadcast news in Albayzin 2010 evaluation campaign. EURASIP J. Audio Speech Music Process. **2012**(19) (2012)
35. Zhang, A., Wang, Q., Zhu, Z., Paisley, J.W., Wang, C.: Fully supervised speaker diarization. In: ICASSP, pp. 6301–6305 (2019)