



Interrelate Training and Searching: A Unified Online Clustering Framework for Speaker Diarization

Yifan Chen^{1,2}, Yifan Guo^{1,2}, Qingxuan Li³, Gaofeng Cheng^{1,†}, Pengyuan Zhang^{1,2}, Yonghong Yan^{1,2}

¹Key Laboratory of Speech Acoustics and Content Understanding, Institute of Acoustics, Chinese Academy of Sciences, China, ²University of Chinese Academy of Sciences, China

³Tsinghua University, Beijing, China

{chenyifan, guoyifan, chenggaofeng, zhangpengyuan, yanyonghong}@hcccl.ioa.ac.cn,
liqx20@mails.tsinghua.edu.cn

Abstract

For online speaker diarization, samples arrive incrementally, and the overall distribution of the samples is invisible. Moreover, in most existing clustering-based methods, the training objective of the embedding extractor is not designed specially for clustering. To improve online speaker diarization performance, we propose a unified online clustering framework, which provides an interactive manner between embedding extractors and clustering algorithms. Specifically, the framework consists of two highly coupled parts: clustering-guided recurrent training (CGRT) and truncated beam searching clustering (TBSC). The CGRT introduces the clustering algorithm into the training process of embedding extractors, which could provide not only cluster-aware information for the embedding extractor, but also crucial parameters for the clustering process afterward. And with these parameters, which contain preliminary information of the metric space, the TBSC penalizes the probability score of each cluster, in order to output more accurate clustering results in online fashion with low latency. With the above innovations, our proposed online clustering system achieves 14.48% DER with collar 0.25 at 2.5s latency on the AISHELL-4, while the DER of the offline agglomerative hierarchical clustering is 14.57%.

1. Introduction

Speaker diarization aims at detecting the speech activities of each person in a conversation. In other words, it is a task of detecting "who speak when". Recent speaker diarization systems are based on two main approaches, i.e. clustering-based approaches and fully supervised approaches [1]. Fully supervised approaches optimize diarization objects directly in a fully supervised way and could achieve promising performance, especially when speech from different speakers is highly overlapped [2, 3]. However, fully supervised approaches need data with accurate timestamp annotations, which are hard to get in many scenarios. Compared with fully supervised approaches, clustering-based methods generate diarization results in an unsupervised way by conducting clustering. And thus, only speaker recognition data are required to train the clustering-based systems, which are much easier to obtain.

However, in most clustering-based methods, the procedures of embedding extracting and clustering are separated during both training and inference, which means the training objective of the embedding extractors could be sub-optimal for clustering, and the structural information of clusters are not introduced into the embedding extractors either. Furthermore, with little prior

information about the metric space of embeddings, it is difficult for clustering algorithms to decide whether an embedding belongs to a new class or an existing one, especially when considering clustering in an online fashion. To solve such problems, we propose a unified online clustering framework which not only optimizes embedding extractor towards online clustering objective, but also provides the clustering algorithm with essential information mentioned above. And our basic idea is that it would be easier for the online clustering algorithm to make a decision when a new sample comes if we could know which samples are definitely from the same class and in which condition we should assign a new sample to a new class.

The contributions of this work could be summarized into three aspects:

(1) We propose a unified training and inference framework for online clustering, which provides an interactive manner between embedding extractors and clustering algorithms.

(2) Under this framework, an embedding extractor training scheme named clustering-guided recurrent training (CGRT) is proposed to introduce the cluster-aware information into the training process of the embedding extractor, and also provide structural information of the embeddings' metric space for the clustering algorithm simultaneously.

(3) An online clustering method based on the truncated beam search (TBSC) is further introduced, which utilizes the information provided by the CGRT to penalize probability scores for potential candidates.

2. Related Work

2.1. Speaker diarization

For a clustering-based system, the typical method is to conduct the hierarchical clustering [4] or the spectral clustering [5] directly on the extracted embeddings such as i-vectors, d-vectors, and x-vectors [6, 7, 8]. However, the direct use of these algorithms does not consider the diarization task's specialty. Therefore, several sophisticated clustering models are designed. Fox et al. [9] introduce the Dirichlet processes mixture model into the diarization task. Diez et al. [10, 11, 12] designed a Bayesian model, i.e. VB-HMM, to conduct clustering, whose hidden states correspond to the speaker identities while the observable states correspond to the embeddings. Besides, graph neural networks are used for clustering by Wang et al. [13], where the structural information of the embeddings is utilized. However, all these methods conduct clustering in an offline way. And since they need knowledge about the whole distribution of the samples, it is difficult to extend these methods to online inference. To alleviate the training-inference mismatch problem

[†] Corresponding Author.

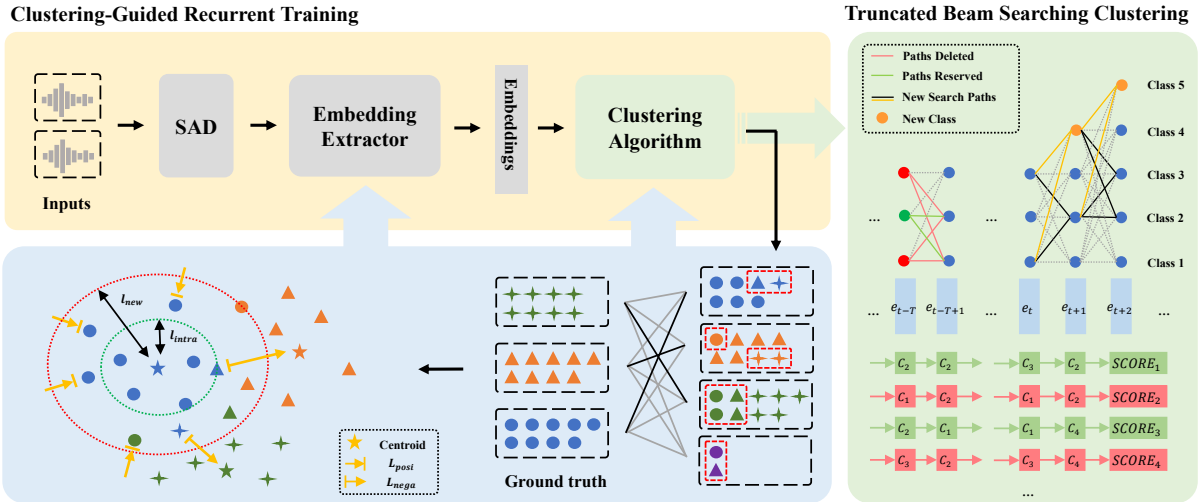


Figure 1: Structure of the proposed unified online clustering for speaker diarization. Clustering-guided recurrent training is shown in the left part, where clustering results are compared with the ground truths during the training process of embedding extractor, trying to introduce cluster-aware information for the embedding extractor. Meanwhile, the results of the comparison could also supply prior information of metric space for clustering algorithm as shown in the right part. And the truncated beam searching clustering tries to find locally optimal combination of embeddings by using the parameters to penalize the probability score of each path and outputs results in online fashion. Note that the color of each sample represents its assigned label, while the shape represents the ground truth.

in the clustering based framework, fully supervised methods are introduced. UIS-RNN [14] uses the recurrent neural networks (RNN) to model the transition probabilities among different speakers, which could output diarization results in an online fashion. EEND [2, 15, 16] discards the use of the speakers’ embeddings and directly optimizes the diarization task in an end-to-end manner. Moreover, Xue et al. [17] introduce a speaker-tracing buffer in order to solve the across-chunk permutation issue when extending the EEND to online applications. Besides, there are other ways to conduct diarization. RPNSD [18] utilizes the Faster R-CNN [19] to conduct the diarization task, which is a commonly used image segmentation approach. And TS-VAD [3] compares the embedding of a chunk of audio with the embedding of a target speaker to decide whether the chunk includes the corresponding speaker or not. All these non-clustering based diarization methods require domain-matched training data with accurate timestamp annotations, which are hard to obtain in some scenarios.

2.2. Metric embeddings learning

Metric embeddings learning has developed rapidly in recent years, aiming to map semantically similar samples into similar embeddings and map different ones into discrepant embeddings. Weinberger et al. [20] propose a loss that could pull the data points from the same class together and push the ones belonging to different classes away from each other. Based on it, and benefiting from the development of deep learning, Triplet loss [21] is proposed and is applied successfully on large-scale face recognition. And Hermans et al. [22] optimize hard samples firstly rather than optimize all samples simultaneously. Besides, in order to enhance the discriminability of embeddings, classification margins are introduced [23, 24]. All these methods are optimized for verification or identification task, which may be sub-optimal for online clustering applications. And motivated by these cutting-edge methods, our clustering-guided recurrent training does not only constrain the metric space of embeddings, but also provides such information for the truncated

beam searching clustering algorithm for clustering afterward.

2.3. Online clustering algorithm

Benefiting from the development of machine learning, several works conduct clustering in an online fashion. Typical methods cluster incremental samples in a greedy manner [25]. However, in this way, thresholds are required to assign a sample to a new class and to merge two classes into one class, which limits the generalization of algorithms in applications. Instead of using a threshold to judge a new sample belongs to which classes, Liu et al. [26] proposes within-cluster dispersion as a criterion for classes selection. And particle filters are introduced to the online application by Mansinghka et al. [27], which use a sequential Monte Carlo scheme to approximate the inference of the Chinese restaurant process mixture model. However, all these methods suffer from the lack of the prior information on the samples’ overall distribution and metric space. Therefore, it is still challenging for them to decide whether a new sample belongs to a new class or an existing one.

3. Our Approach

As shown in Fig. 1, our unified online clustering framework consists of two highly coupled parts: an embedding extractor training scheme, i.e. clustering-guided recurrent training (CGRT), and an online clustering algorithm, i.e. truncated beam searching clustering (TBSC). Clustering-guided recurrent training aims to optimize embedding extractor towards online clustering object, and simultaneously provide prior information of metric space for the clustering algorithm. TBSC tries to conduct clustering and output results in an online fashion with the assistance of these prior information. More details about CGRT and TBSC are as follows.

We will use the following notations to introduce our method. Assume $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^N$ is a set of training samples, and the corresponding speaker labels is $\{y_i\}_{i=1}^N$. Embedding extractor $\{\mathcal{F}_t\}_{t=1}^T$ which is trained after t iterations, maps train-

ing samples $\{\mathbf{x}_i\}_{i=1}^N$ to embeddings $\{e_i^t\}_{i=1}^N$. The embeddings are clustered into K clusters, $\{(\mathcal{C}_i, \boldsymbol{\mu}_i)\}_{i=1}^K$, where \mathcal{C}_i represents embedding set, and $\boldsymbol{\mu}_i$ is the corresponding center.

3.1. Clustering-Guided Recurrent Training

In order to optimize embedding extractor towards clustering object and provide practical information for clustering algorithm, it is essential to know which samples will lead to clustering mistakes. Therefore, the basic idea is to conduct clustering and embedding extractor training alternatively in a prediction-correction way. Specifically, in prediction stage, the embedding extractor provides intermediate embeddings to the clustering algorithm and obtains clustering results. In correction stage, clustering results are compared with ground truths, thus providing feedback to embedding extractor and clustering algorithm, i.e. guiding the coming training process of embedding extractor and updating parameters containing information of metric space for the clustering algorithm.

Algorithm 1: Clustering-Guided Recurrent Training.

Input: embedding extractor \mathcal{F} , clustering algorithm \mathcal{H} , training set \mathcal{X}

- 1 **for** $t = 1, 2, \dots, T$ **do**
- 2 **// prediction stage**
- 3 Sampling an subset $\mathcal{X}'_t = \{(\mathbf{x}_i, y_i)\}_{i=1}^M$
- 4 Calculate embeddings $\{e_i^t\}_{i=1}^M$ using \mathcal{F}_{t-1}
- 5 **// correction stage**
- 6 Cluster $\{e_i^t\}_{i=1}^M$ and get l_{intra}, l_{new}
- 7 Calculate $\mathcal{L}_{posi}, \mathcal{L}_{nega}$ with $l_{intra}, l_{new}, \{e_i^t\}_{i=1}^M$
- 8 Conduct optimization with $\mathcal{L}_{posi}, \mathcal{L}_{nega}$ to get \mathcal{F}_t
- 9 **end**

We now describe more specific steps of our proposed training scheme. As shown in algorithm 1, CGRT consists of the following steps. Suppose that we have a pre-trained feature extractor \mathcal{F}_0 . In prediction stage, a subset $\mathcal{X}'_t \subset \mathcal{X}$ is sampling from the whole training set randomly. And embedding extractor \mathcal{F}_{t-1} provides intermediate embedding set $\{e_i^t\}$ corresponding to \mathcal{X}'_t . Then, we conduct clustering on these embeddings and obtain clustering results $\{\mathcal{C}_i\}$.

In correction stage, we need to distinguish "positive" embeddings and "negative" embeddings in a cluster. To this end, we firstly find the perfect matching between obtained clusters and ground truths and assign each embedding a "positive" or "negative" label. Assume K is the number of the clusters, $\{\mathcal{Y}_j\}_{j=1}^K$ are the speaker label sets generated by the clustering, and $\{\mathcal{G}_i\}_{i=1}^L$ are sets that consist of real speaker labels of samples (ground truths). L is the real number of speakers. Then we can get a perfect matching using the KM algorithm with a bipartite graph defined by $W_{i,j} = \frac{|\mathcal{G}_i \cap \mathcal{Y}_j|}{|\mathcal{G}_i \cup \mathcal{Y}_j|} * |\mathcal{Y}_j|$, and thus each embedding e_i could be assigned to the "positive" set \mathcal{E}_{pos} or the "negative" set \mathcal{E}_{neg} .

Next, we calculate two hyperparameters that contain practical information for both clustering algorithm and embedding extractor optimizing algorithm, i.e., l_{intra} and l_{new} :

$$l_{intra} = \min_{e_i \in \mathcal{E}_{neg}} \{d(\boldsymbol{\mu}_{false}, e_i)\} \quad (1)$$

$$l_{new} = \max_{e_i \in \mathcal{E}_{pos}} \{d(\boldsymbol{\mu}_{true}, e_i)\} \quad (2)$$

where $d(\mathbf{x}, \mathbf{y})$ means the cosine distance between two vectors. The meaning of l_{intra} is the minimum distance of negative em-

beddings from centers of false classes. And below this threshold, the embeddings should belongs to the existing class. The meaning of l_{new} is the maximum distance of embeddings from centers of true classes. The embeddings should come from a new class when all distances from existing centers of classes are larger than it.

Using these hyperparameters that include information about metric space, the embedding extractor could be optimized with the following two losses:

$$\mathcal{L}_{posi} = \frac{1}{N} \sum_{i,j, y_i=y_j} \max\{d(e_i, e_j) - l_{intra}, 0\} \quad (3)$$

$$\mathcal{L}_{nega} = \frac{1}{N} \sum_{e_i \in \mathcal{E}_{neg}} \max\{d(\boldsymbol{\mu}_{true}, e_i) - d(\boldsymbol{\mu}_{false}, e_i) + l_{new}, 0\} \quad (4)$$

where \mathcal{L}_{posi} aims at pulling samples which belong to the same class together. And \mathcal{L}_{nega} targets at pushing negative data points from the false center $\boldsymbol{\mu}_{false}$ further away, and pulling them towards the center $\boldsymbol{\mu}_{true}$ they belong to.

3.2. Truncated Beam Searching Clustering

For online clustering applications, it is not easy to decide whether a new sample belongs to a new cluster or an existing one since samples arrive incrementally. To address this problem, on the one hand, clustering-guided recurrent training is proposed above, which could not only constrain embeddings' distribution in metric space, but also provides hyperparameters containing information about metric space. On the other hand, a clustering method using a heuristic search algorithm is proposed as a fault-tolerant design for online applications.

The more specific steps of truncated beam searching clustering are as follows. It requires two essential hyperparameters: beam size B and latency T_0 . The beam size determines the number of search paths preserved during search process. And latency T_0 determines truncated length of each search route, thus determining the latency between inputs and outputs. When a new sample arrives, the probabilities of all possible classes are calculated. The details are shown in algorithm 2.

Algorithm 2: Truncated Beam Searching Clustering.

Input: Beam size B , Latency T_0 , Embedding sequence $\{e_i, i \in [1, T], l_{intra}, l_{new}\}$

- 1 Initialize a max-queue (paths set) $Q = \emptyset$
- 2 **for** $t = 1, 2, \dots, T$ **do**
- 3 Calculate scores of assigning e_t to a new class: $s(C_{new})$ and existing classes: $s(C_i)$.
- 4 Use these scores to extend paths to the candidate classes of e_t respectively.
- 5 Output the class label at $t - T_0$ corresponding to the current max-score paths as $result_{t-T_0}$.
- 6 Delete paths not containing $result_{t-T_0}$.
- 7 Truncate the history of reserved paths before $t - T_0$.
- 8 **if** $|Q| > B$ **then**
- 9 Shrink Q by score.
- 10 **end**
- 11 **end**

In the score calculating step in algorithm 2, we add some constraints to our search path. If all distances between centers of clusters and the new sample are larger than l_{new} , there is a great possibility that the new sample comes from a new cluster. Therefore, a high score will be assigned to the path which leads

to a new cluster. And if the distance between a new sample and the center of a cluster is smaller than l_{intra} , it would be potential that the new sample belongs to this cluster. Besides, we penalize the continuity of label sequence based on the fact that, in most cases, the current frame and the previous frame belongs the same class.

$$s(C_{new}) = \begin{cases} S_0, & \min_j \{d(\mathbf{e}_t, \boldsymbol{\mu}_j)\} \geq l_{new} \\ f(\min_j \{d(\mathbf{e}_t, \boldsymbol{\mu}_j)\}), & \min_j \{d(\mathbf{e}_t, \boldsymbol{\mu}_j)\} < l_{new} \end{cases} \quad (5)$$

$$s(C_i) = \begin{cases} S_1 + \lambda \cdot \delta_{C^{t-1}, C_i}, & d(\mathbf{e}_t, \boldsymbol{\mu}_i) \leq l_{intra} \\ g(d(\mathbf{e}_t, \boldsymbol{\mu}_i)) + \lambda \cdot \delta_{C^{t-1}, C_i}, & d(\mathbf{e}_t, \boldsymbol{\mu}_i) > l_{intra} \end{cases} \quad (6)$$

Note that the parameters of truncated beam searching clustering for inference could use the last ones in the training stage or use the ones generated on the eval set if available.

4. Experiments

4.1. Datasets

We use open-source datasets to validate our proposed method, i.e. CN-Celeb1&2 [28, 29] and AISHELL-4 [30]. CN-Celeb1&2 [28, 29] are two large-scale Chinese speaker recognition corpora. CN-Celeb1 [28] consists of 274 hours audio from 1000 speakers, while CN-Celeb2 [29] contains 1090 hours audio from 2000 speakers. We use these datasets to train the embedding extractor. For speaker diarization task, AISHELL-4 [30] dataset is adopted, which is an open-source meeting speech recognition datasets containing accurate timestamps. And the number of speakers in each session is 4 to 8. Note that we only use the test set of AISHELL-4 [30].

4.2. Implementation details

For implementation details, an energy-based voice activity detector (VAD) implemented by kald [31] is adopted to remove silence. ResNet-50 with two fully connected layers is employed to conduct the speaker classification task, and the raw waveform is split into 4s pieces as input. During clustering-guided recurrent training, in each circulation, audios are sampling from 4-8 speakers. And we firstly adopt spectral clustering to provide initial results. Then, TBSC with beam size 1 is employed in order to speed up the CGRT process. And we adopt cosine distance to conduct optimizing procedures. For the clustering task, embeddings are extracted every 100ms, and the length of the windows is 1s. In equation 5 and 6, we adopt $f(x) = \log(x)$, $g(x) = \log(1 - x)$, $S_0 = S_1 = 0$.

4.3. Results

Table 1: Speaker diarization result on AISHELL-4.

Mode	Methods	DER	
		collar 0.25	collar 0
Offline	X-Vector + AHC	14.57	23.01
	X-Vector + SC	14.31	22.53
	X-Vector + VB-HMM [11]	11.10	16.76
Online	X-Vector + LFC	17.66	23.21
	X-Vector + DBC [26]	17.01	22.84
Online	X-Vector + CGRT + LFC	16.41	22.13
	X-Vector + CGRT + TBSC	14.48	21.01

We carry out experiments to validate the effectiveness of the

proposed framework. Firstly, we use X-Vector with PLDA similarity measurement and agglomerative hierarchical clustering (AHC) as an offline baseline. And x-vector with spectral clustering (SC) is also adopted to make a comparison. Secondly, in order to validate our proposed method in the online setting, online clustering methods, i.e. leader-follower clustering (LFC) and dispersion-based clustering (DBC) [26] are also compared. Leader-follower clustering compares a new embedding with all clusters' centroid and assign it to the most similar one or making a new clutter, which could be seen as the following degenerate situation of TBSC: beam size $B = 1$, latency $T = 0$, and l_{new}, l_{intra} are not used. The result suggests that with the help of CGRT and TBSC, our online framework could get close to offline clustering. And our approach could outperform previous online clustering methods.

4.4. Analysis of parameters

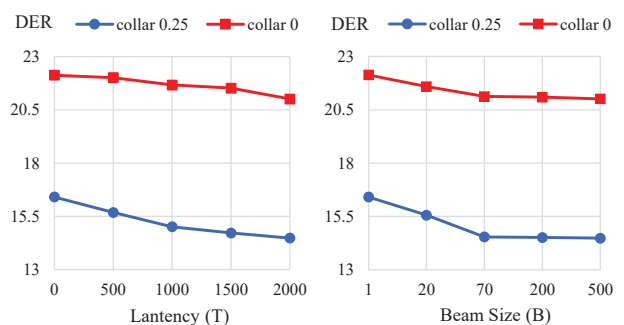


Figure 2: Impact of the beam size B and latency T in TBSC.

To analyze the effectiveness of our system, we conduct more experiments on the AISHELL-4 [30] dataset. Latency T_0 and Beam size B in TBSC are adjusted to test performance. The results show that the tradeoff between latency and accuracy exists. And with more memory consumption and latency, TBSC could provide more accurate clustering results.

5. Conclusion

In this paper, we propose a unified training and inference framework for online clustering. The proposed framework consists of clustering-guided recurrent training (CGRT) and truncated beam searching clustering (TBSC), which provides an interactive manner between embedding extractor and clustering algorithm. Since CGRT introduces the cluster-aware information into the embedding extractor training process, and provides prior information about embeddings' metric space to the clustering algorithm simultaneously, TBSC could conduct online clustering with the knowledge of the coming samples in a matched way. With a robust embedding extractor that could initially extract accurate x-vectors, our method achieves competitive results on the diarization task in an online fashion.

6. Acknowledgment

This work was partially supported by the Youth Innovation Promotion Association, Chinese Academy of Sciences and the Frontier Exploration Project Independently Deployed by Institute of Acoustics, Chinese Academy of Sciences under Grant QYTS202011.

7. References

- [1] X. Anguera, S. Bozonnet, N. Evans, C. Fredouille, G. Friedland, and O. Vinyals, "Speaker diarization: A review of recent research," *IEEE Transactions on audio, speech, and language processing*, vol. 20, no. 2, pp. 356–370, 2012. [1](#)
- [2] Y. Fujita, N. Kanda, S. Horiguchi, Y. Xue, K. Nagamatsu, and S. Watanabe, "End-to-end neural speaker diarization with self-attention," in *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2019, pp. 296–303. [1](#), [2](#)
- [3] I. Medennikov, M. Korenevsky, T. Prisyach, Y. Khokhlov, M. Korenevskaya, I. Sorokin, T. Timofeeva, A. Mitrofanov, A. Andrusenko, I. Podluzhny *et al.*, "Target-speaker voice activity detection: A novel approach for multi-speaker diarization in a dinner party scenario," *Proc. Interspeech 2020*, pp. 274–278, 2020. [1](#), [2](#)
- [4] S. C. Johnson, "Hierarchical clustering schemes," *Psychometrika*, vol. 32, no. 3, pp. 241–254, 1967. [1](#)
- [5] A. Ng, M. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," *Advances in neural information processing systems*, vol. 14, 2001. [1](#)
- [6] N. Dehak, P. J. 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, 2010. [1](#)
- [7] E. Variansi, X. Lei, E. McDermott, I. L. Moreno, and J. Gonzalez-Dominguez, "Deep neural networks for small footprint text-dependent speaker verification," in *2014 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2014, pp. 4052–4056. [1](#)
- [8] 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, pp. 5329–5333. [1](#)
- [9] E. B. Fox, E. B. Sudderth, M. I. Jordan, and A. S. Willsky, "The sticky hdp-hmm: Bayesian nonparametric hidden markov models with persistent states," *Arxiv preprint*, 2007. [1](#)
- [10] M. Diez, L. Burget, F. Landini, and J. Černocký, "Analysis of speaker diarization based on bayesian hmm with eigenvoice priors," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 355–368, 2019. [1](#)
- [11] M. Diez, L. Burget, S. Wang, J. Rohdin, and J. Černocký, "Bayesian hmm based x-vector clustering for speaker diarization," in *Interspeech*, 2019, pp. 346–350. [1](#), [4](#)
- [12] F. Landini, J. Profant, M. Diez, and L. Burget, "Bayesian hmm clustering of x-vector sequences (vbx) in speaker diarization: theory, implementation and analysis on standard tasks," *Computer Speech & Language*, vol. 71, p. 101254, 2022. [1](#)
- [13] J. Wang, X. Xiao, J. Wu, R. Ramamurthy, F. Rudzicz, and M. Brudno, "Speaker attribution with voice profiles by graph-based semi-supervised learning," *Proc. Interspeech 2020*, pp. 289–293, 2020. [1](#)
- [14] A. Zhang, Q. Wang, Z. Zhu, J. Paisley, and C. Wang, "Fully supervised speaker diarization," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 6301–6305. [2](#)
- [15] S. Horiguchi, Y. Fujita, S. Watanabe, Y. Xue, and K. Nagamatsu, "End-to-end speaker diarization for an unknown number of speakers with encoder-decoder based attractors," *Proc. Interspeech 2020*, pp. 269–273, 2020. [2](#)
- [16] S. Maiti, H. Erdogan, K. Wilson, S. Wisdom, S. Watanabe, and J. R. Hershey, "End-to-end diarization for variable number of speakers with local-global networks and discriminative speaker embeddings," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 7183–7187. [2](#)
- [17] Y. Xue, S. Horiguchi, Y. Fujita, S. Watanabe, P. García, and K. Nagamatsu, "Online end-to-end neural diarization with speaker-tracing buffer," in *2021 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2021, pp. 841–848. [2](#)
- [18] Z. Huang, S. Watanabe, Y. Fujita, P. García, Y. Shao, D. Povey, and S. Khudanpur, "Speaker diarization with region proposal network," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6514–6518. [2](#)
- [19] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *Advances in neural information processing systems*, vol. 28, 2015. [2](#)
- [20] K. Q. Weinberger, J. Blitzer, and L. Saul, "Distance metric learning for large margin nearest neighbor classification," *Advances in neural information processing systems*, vol. 18, 2005. [2](#)
- [21] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 815–823. [2](#)
- [22] A. Hermans, L. Beyer, and B. Leibe, "In defense of the triplet loss for person re-identification," *arXiv preprint arXiv:1703.07737*, 2017. [2](#)
- [23] F. Wang, J. Cheng, W. Liu, and H. Liu, "Additive margin softmax for face verification," *IEEE Signal Processing Letters*, vol. 25, no. 7, pp. 926–930, 2018. [2](#)
- [24] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "Arcface: Additive angular margin loss for deep face recognition," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 4690–4699. [2](#)
- [25] M. Charikar, C. Chekuri, T. Feder, and R. Motwani, "Incremental clustering and dynamic information retrieval," *SIAM Journal on Computing*, vol. 33, no. 6, pp. 1417–1440, 2004. [2](#)
- [26] D. Lilt and F. Kubala, "Online speaker clustering," in *2004 IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 1. IEEE, 2004, pp. I-333. [2](#), [4](#)
- [27] V. K. Mansinghka, D. M. Roy, R. Rifkin, and J. Tenenbaum, "Aclass: An online algorithm for generative classification," in *Proceedings of the 11th International Conference on Artificial Intelligence and Statistics*, vol. 2, 2007, pp. 315–322. [2](#)
- [28] Y. Fan, J. Kang, L. Li, K. Li, H. Chen, S. Cheng, P. Zhang, Z. Zhou, Y. Cai, and D. Wang, "Cn-celeb: a challenging chinese speaker recognition dataset," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 7604–7608. [4](#)
- [29] L. Li, R. Liu, J. Kang, Y. Fan, H. Cui, Y. Cai, R. Vippera, T. F. Zheng, and D. Wang, "Cn-celeb: multi-genre speaker recognition," *Speech Communication*, 2022. [4](#)
- [30] Y. Fu, L. Cheng, S. Lv, Y. Jv, Y. Kong, Z. Chen, Y. Hu, L. Xie, J. Wu, H. Bu *et al.*, "Aishell-4: An open source dataset for speech enhancement, separation, recognition and speaker diarization in conference scenario," *arXiv preprint arXiv:2104.03603*, 2021. [4](#)
- [31] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz *et al.*, "The kaldı speech recognition toolkit," in *IEEE 2011 workshop on automatic speech recognition and understanding*, no. CONF. IEEE Signal Processing Society, 2011. [4](#)