COMMUNITY DETECTION GRAPH CONVOLUTIONAL NETWORK FOR OVERLAP-AWARE SPEAKER DIARIZATION

*Jie Wang[†1], Zhicong Chen[†1], Haodong Zhou[1], Lin Li[*1], Qingyang Hong[2]*

[1]School of Electronic Science and Engineering, Xiamen University, China
[2]School of Informatics, Xiamen University, China
lilin@xmu.edu.cn

ABSTRACT

The clustering algorithm plays a crucial role in speaker diarization systems. However, traditional clustering algorithms suffer from the complex distribution of speaker embeddings and lack of digging potential relationships between speakers in a session. We propose a novel graph-based clustering approach called Community Detection Graph Convolutional Network (CDGCN) to improve the performance of the speaker diarization system. The CDGCN-based clustering method consists of graph generation, sub-graph detection, and Graph-based Overlapped Speech Detection (Graph-OSD). Firstly, the graph generation refines the local linkages among speech segments. Secondly the sub-graph detection finds the optimal global partition of the speaker graph. Finally, we view speaker clustering for overlap-aware speaker diarization as an overlapped community detection task and design a Graph-OSD component to output overlap-aware labels. By capturing local and global information, the speaker diarization system with CDGCN clustering outperforms the traditional Clustering-based Speaker Diarization (CSD) systems on the DIHARD III corpus.

Index Terms— speaker diarization, graph convolutional network, speaker clustering, community detection

1. INTRODUCTION

Speaker diarization is a problem of grouping speech segments in an audio recording according to the speakers' identities. We have witnessed the rising popularity of speaker diarization over recent years for its significant applications of minutes of meetings, multi-speaker transcription, pre-processing for automatic speech recognition (ASR) [1][2], and so on. As the deployments for scenarios have grown in complexity, speaker diarization systems confront many difficulties, such as the unknown number of speakers and handling the overlapped speech.

Clustering-based approaches are widely used in speaker diarization because it allows for flexible and scalable speaker modeling using various techniques [3][4][5]. There are three modules in clustering-based Speaker Diarization (CSD) systems: speaker embedding extractor, clustering module, and post-processing module. Typically, the clustering modules in CSD systems utilize conventional clustering algorithms, such as Agglomerative Hierarchical Clustering (AHC) [6][7], Spectral Clustering (SC) [8][9][10] and K-means[11], to perform speaker embeddings clustering. However,

---

Fig. 1: An illustration of the speaker diarization system pipeline with the CDGCN clustering method. The Rich Transcription Time Marked (RTTM) is the output of the speaker diarization systems.

traditional conventional clustering algorithms suffer from complicated distribution of speaker embeddings [12] and is sensitive to hyper-parameter. For example, SC assumes the sizes of clusters are relatively balanced, while K-means assumes the clusters are spherical. Moreover, the performance of AHC is affected by the threshold sensitively. These assumptions limit the speaker clustering performance and degrade the diarization quality.

The distribution of speakers is hard to be modeled with Euclidean structures, because of the complex interrelation among speakers. Graph Convolutional Network (GCN) [13] is proposed to handle the data of non-Euclidean structure. Many GCN-based clustering methods are recently proposed for large-scale embeddings clustering instead of relying on hand-crafted criteria. Tong et al. [14] adopted Detection Segmentation Graph Convolutional Network (DSGCN) for semi-supervised speaker recognition. Wang et al. [15] used a GCN to refine speaker embeddings for affinity matrix on speaker diarization system.

Inspired by these works, we proposed a new GCN-based clustering approach with community detection for speaker diarization named Community Detection Graph Convolutional Network (CDGCN). We regard the clustering of speaker embeddings as a speaker graph generation and sub-graph detection task. The key idea is to build a refined speaker graph for segment embeddings and globally partition speaker graph to assign speaker labels for segments. The CDGCN-based clustering method also can assign multi-labels for each node to handle overlapped speech.

The remainder of this paper is organized as follows. In Section 2, we revisit graph convolutional networks. The proposed approach is addressed in Section 3. In Section 4.1, we describe the dataset of our experiments. In Section 5, we evaluate the proposed systems on the DIHARD III [16]. Finally Section 6 concludes this work.

2. GRAPH CONVOLUTIONAL NETWORK

In our work, a modified GCN [17] model was adopted to build speaker graphs. The input of the GCN model is an embedding matrix $\boldsymbol{H} \in \mathbb{R}^{K \times D}$ together with an adjacency matrix $\boldsymbol{A} \in \mathbb{R}^{K \times K}$, where K is the number of nodes in a graph and D is the dimension of the embeddings. The feedforward of the GCN model can be summarized in two steps:

(1) **Aggregation**: The aggregation processing allows each node to learn the information from neighbors on the graph. After graph aggregation, the GCN layer transforms $\boldsymbol{H}^{(l)}$ into a hidden feature matrix $\boldsymbol{H}^{(l+1)}$. The aggregation is formulated as follows:

$$\boldsymbol{H}^{(l+1)} = \sigma([\boldsymbol{H}^{(l)} \parallel \hat{\boldsymbol{A}} \boldsymbol{H}^{(l)}] \boldsymbol{W}^{(l)}) \quad (1)$$

where $\boldsymbol{H}^{(l)} \in \mathbb{R}^{K \times D^{(l)}}$, $\boldsymbol{H}^{(l+1)} \in \mathbb{R}^{K \times D^{(l+1)}}$ denotes the output data with $D^{(l+1)}$ dimensions in $(l+1)$-th layer, σ is the Relu activation function, $\boldsymbol{W}^{(l)} \in \mathbb{R}^{2D^{(l)} \times D^{(l+1)}}$ is a learnable weight matrix in the l-th layer, $\hat{\boldsymbol{A}}$ is the normalized and regularized affinity matrix with $K \times K$ size and each row is summed up to 1. "$\parallel$" denotes matrix concatenation operation along the feature dimension. The normalized affinity matrix $\hat{\boldsymbol{A}}$ is formulated as:

$$\hat{\boldsymbol{A}} = \widetilde{\boldsymbol{D}}^{-\frac{1}{2}} \widetilde{\boldsymbol{A}} \widetilde{\boldsymbol{D}}^{-\frac{1}{2}} \quad (2)$$

where, $\widetilde{\boldsymbol{A}} = \boldsymbol{A} + \boldsymbol{I}$ is the adjacency matrix with self connection, $\boldsymbol{I}$ is the unit matrix and $\widetilde{\boldsymbol{D}}$ denotes the degree matrix of $\widetilde{\boldsymbol{A}}$ with $\widetilde{\boldsymbol{D}}_{ii} = \sum_j \widetilde{\boldsymbol{A}}_{ij}$.

(2) **Prediction**: Finally, the prediction labels of nodes $\boldsymbol{Y} = \{y_1, y_2, ..., y_K\} \in \mathbb{R}^K$ are generated by two stacked linear layers with a softmax function. The labeling principle is that $y_k=1$ if there is a linkage between the pivot node and the k-th node; otherwise reverse. The GCN is trained by Binary Cross Entropy (BCE) loss.

3. PROPOSED APPROACH

3.1. System pipeline

The CDGCN-based speaker diarization system pipeline is shown in Figure 1. Firstly, input samples are split into segments with slide windows. Then, the embedding extractor converts speech segments into fixed dimension vectors called x-vectors $\boldsymbol{X} \in \mathbb{R}^{N \times D}$ where N is the number of segments and D is the feature dimension of an embedding. We adopt a ResNet-34-SE model to build the extractor. After that, we construct the raw speaker graph by calculating cosine similarity scores between embeddings. The CDGCN-based clustering module takes the raw speaker graph and outputs overlap-aware speaker labels. The diarization results follow from the labels.

3.2. CDGCN-based clustering

The overall block diagram of CDGCN is shown in Figure 2. The basic concept of the CDGCN is to estimate the topological connection of speech segments and use a community detection algorithm to find the optimal partitions. CDGCN-based clustering method contains graph generation, sub-graphs detection, and graph-based overlapped speech detection (Graph-OSD). Each component of CDGCN will be described as follows.

3.2.1. Graph Generation

The input of clustering module is a raw graph $\mathcal{G} = (\mathcal{V}, \boldsymbol{\mathcal{E}})$, where nodes $\mathcal{V} = \{v_1, v_2, ..., v_N\} \in \mathbb{R}^N$ represent speech segments, edges $\boldsymbol{\mathcal{E}} = \{\boldsymbol{e}_1, \boldsymbol{e}_2, ..., \boldsymbol{e}_N\} \in \mathbb{R}^{N \times N}$ are cosine similarity scores

between pairs of embeddings and N is the number of segments. The raw graph is a complex full-connected graph which is vulnerable to noise. In order to tackle this problem, we design a graph generation to refine interrelations between speech segments according to local context information. Firstly, we adopt the K-Nearest Neighbors (KNN) algorithm to create sub-graphs for each node. The sub-graph $\mathcal{G}_n = (\mathcal{V}_n, \mathcal{E}_n)$ is built for n-th pivot node, where $\mathcal{V}_n \in \mathbb{R}^K$ is the top-K nearest neighbor of pivot node and $\mathcal{E}_n \in \mathbb{R}^K$ denotes the similarity among n-th pivot node and its neighbors. For example, as shown in Figure 2 (a), let K and the nodes number N be 6 and 12 respectively. The raw speaker sub-graphs $\mathcal{G}_n$ are fed into GCN model mentioned in Section 2 and the refined sub-graphs $\hat{\mathcal{G}}_n = (\mathcal{V}_n, \hat{\mathcal{E}}_n)$ are predicted, where $\hat{\mathcal{E}}_n = \{\hat{e}_n^1, \hat{e}_n^2, ..., \hat{e}_n^K\} \in \mathbb{R}^K$ are predicted edges and $\hat{e}_n^K$ indicates the probability that pivot node and k-th node belong to the same cluster. Then, the refined speaker sub-graphs are merged to acquire the total refined speaker graph $\hat{\mathcal{G}} = (\mathcal{V}, \hat{\mathcal{E}})$ which is a weighted undirected graph. In the graph merging stage, multiple edges between two nodes keep the bigger one.

3.2.2. Sub-graphs Detection

One of the main obstacles is to partition the refined graph robustly. The sub-graphs detection predicts the most-likely community label of nodes on the speaker graph. Zheng et al. [18] use Leiden community detection[19] with Uniform Manifold Approximation and Projection (UMAP) for speaker clustering on the simulated meetings. In this part, we adopt Leiden community detection for sub-graphs detection. Community is interpreted as clusters of densely interconnected nodes that are only sparsely connected with the rest on the graph [20]. In our work, each community label corresponds to a speaker label. Community detection aims to group nodes with an optimization quality function. The higher the Q is, the better the clustering result we may obtain. The quality function Q [19] is represented as:

$$Q = \sum_c \left(m_c - \gamma \frac{K_c^2}{4m} \right) \quad (3)$$

where m_c is total internal edge weight of community c, m is the total number of edges, K_c is the total weighted degree of nodes in community c, and γ is a resolution parameter that controls the number of communities. K_c is given by

$$K_c = \sum_{i|\sigma_i=c} k_i \quad (4)$$

here, σ_i denotes the community label of node i, and k_i is the weighted degree of node i.

The Leiden community detection consists of the following phases:

(1) **Initial partition**: The Leiden algorithm assigns each node to a singleton community.

(2) **Nodes Local moving**: The individual node is moved from one community to another to find a better partition P with higher Q.

(3) **Partition refinement**: In the refinement phase, the refined partition $P_{refined}$ is initially set to a singleton partition. And then, the nodes in each community are merged locally to refine partition $P_{refined}$. After performing refinement, communities in P may be split into subcommunities.

(4) **Graph aggregation**: An aggregation graph is constructed based on $P_{refined}$. In this phase, the node belonging to the same community are merged into a new node.

(5) **Iteration**: Phases 2-4 are repeated until no further improvements of quality function can be made.

Fig. 2: The architecture of Community Detection Graph Convolutional Network based clustering algorithm. "Spk^{1st}" denotes the most-likely speaker labels of nodes, and "Spk^{2nd}" indicates the second most-likely speaker labels of nodes.

3.2.3. Graph-OSD

Overlapped speech handling is the critical processing of speaker diarization. In this work, we propose a Graph-based Overlapped Speech Detection (Graph-OSD) module in CDGCN algorithm. As shown in Figure 2 (c), we view the speaker clustering of diarization as overlapped community detection task. The Graph-OSD is a two-stage model to handle overlapped speech and assume there are at most two speakers at once.

In the first stage, we predict the second community label for each node. According to the refined graph, and the partition created by sub-graphs detection, we calculate the belonging coefficient $b_{(c,i)}$ for each node i, where $b_{(c,i)}$ presents the strength of membership that i-th node belongs to community c. This process is defined as:

$$b_{(c,i)} = \sum_{j|\sigma_j = c} e_{ij} \tag{5}$$

here, σ_j denotes the community label of node j, and e_{ij} is the weighted edge between node i and node j from refined graph. Based on the most-likely community label and belonging coefficient, the second most-likely community $\widetilde{c}_i$ of node i is given by

$$\tilde{c}_i = \underset{c \in C, c \neq \hat{c}_i}{\arg\max} \, b_{(c,i)} \tag{6}$$

where C is the estimated communities number and $\hat{c}_i$ is the most-likely community label.

The next stage predicts the overlapped speech regions and ignores the second speaker labels at non-overlapped regions. We perform an LSTM-based OSD model described in [21] to predict the frame-level overlapped/non-overlapped regions of speech. The model is an end-to-end overlapped speech detection whose output is a frame-level binary sequence, and is trained with the binary cross entropy loss function. Finally, we output the two most likely speakers for each frame in overlapped speech region.

4. DATASETS AND EXPERIMENTAL SETUP

4.1. Data preparation

We evaluate our speaker diarization systems on the DIHARD III corpus. The DIHARD III contains the development (DEV) and evaluation (EVAL) set from 11 domains exhibiting wide variation in equipment. The overlap ratio of DIHARDIII Core and Full dataset is 8.75% and 9.35%, respectively. The detailed training sets of different modules on our speaker diarization systems are described as follows.

- Speaker embedding extractor: We train the embedding extractor with the VoxCeleb2 dataset. The VoxCeleb2 contains over 1 million utterances from 5,994 speakers.

- GCN: We extracted 256-dimensional embeddings for VoxCeleb2. We constructed the sub-graph for each utterance. Each sub-graph is a training instance of the GCN model.

- LSTM-based OSD: We adopted the DIHARD III DEV to train the OSD module.

4.2. Experimental setup

During training, we extracted the 81-dimensional log-mel filter-bank (FBank) with a window size of 25ms and a 10ms shift. In our diarization systems, we split the audio into 1.5s length segments with 0.75s windows shift, and extracted the embeddings of segments with the ResNet-34-SE model from ASV-Subtools[22]. In the GCN module, we stacked four GCN layers and set the K of KNN to 300. The resolution γ of Leiden community detection module is set to 0.6 and the threshold of AHC is set to 0.17.

5. EXPERIMENTAL RESULTS

5.1. Speaker clustering methods

The first experiment explores the performance of different clustering algorithms on speaker diarization systems. The official baseline system provided by DIHARD III [16] consists of Time Delay Neural Network (TDNN) based x-vector extractor, Agglomerative Hierarchical Clustering (AHC) module, and Variational Bayes hidden Markov (VB) re-segmentation module. Our system pipeline is mentioned in Section 3.1, and the difference among S1~S5 is the clustering method. For Systems S1~S3, we respectively performed AHC, K-means, and NME-SC (Normalized Maximum Eigengap Spectral Clustering)[26] as clustering methods. In particular, we adopted Normalized Maximum Eigengap (NME)[26] method to estimate the number of speakers for K-means. We performed the Leiden community detection algorithm on the CDGCN clustering method for system S4 and system S5. In order to investigate the effectiveness of the Graph-OSD module, we removed the module on system S4. In

Table 1: The comparison among different CSD systems on DIHARD III with 0ms collar condition. We evaluated the diarization systems are evaluated on core and full datasets with oracle Voice Activity Detection (VAD). The core is a subset of the full evaluation set and strives for balance cross-domains. DOVER-Lap [23] is a subsystems fusion algorithm.

| ID | Methods | DER(%) | | | |
| | | DEV | | EVAL | |
		Core	Full	Core	Full
Official Baseline[16]	TDNN+AHC	21.05	20.71	21.66	20.75
	TDNN+AHC+VB	20.25	19.41	20.65	19.25
Recent Works	ResNet+SC[24]	16.63	16.51	16.56	15.79
	ResNet+VBx[24]	16.66	16.26	16.67	15.74
	TDNN+VBx w/ OSD[25]	**14.88**	13.87	18.20	15.65
	Res2Net+VBx w/ OSD (DOVER-Lap)[25]	15.18	14.04	18.47	15.81
S1	ResNet+AHC	19.31	19.94	19.27	18.90
S2	ResNet+K-means	25.34	23.05	23.71	21.24
S3	ResNet+NME-SC	18.56	17.89	17.98	16.81
S4	ResNet+CDGCN w/o Graph-OSD(ours)	17.10	16.43	16.50	15.38
S5	ResNet+CDGCN(ours)	15.40	**13.67**	**15.97**	**13.72**

Table 2: Ablation study on CDGCN-based speaker diarization system. + here denotes stacking our components of CDGCN. Oracle OSD indicates that the Graph-OSD replaces the overlapped speech label predicted by the LSTM model with ground truth labels.

| ID | Method | DER(%) | | | |
| | | DEV | | EVAL | |
		Core	Full	Core	Full
S6	Raw-Leiden	24.92	22.03	25.18	21.59
S7	+KNN Graph	18.57	17.70	18.58	17.04
S4	++GCN refinement	17.10	16.43	16.50	15.38
S5	+++Graph-OSD	**15.40**	**13.67**	**15.97**	**13.72**
S8	++++Oracle OSD	11.09	8.94	11.48	8.94

those systems, we tuned the hyper-parameters, including the threshold of AHC and resolution of CDGCN on DIHARD III DEV.

The experimental results are shown in Table 1. We evaluated our systems under the same conditions as recent works [16][24][25]. By comparing the systems S1~S4, the experimental results showed that CDGCN assigned most-likely speaker labels for segments more accurately than other clustering algorithms. The results from system S5 demonstrated that the Graph-OSD module achieved better handling of overlapped speech.

5.2. Ablation experiment

We designed the second experiment to investigate the contribution of each module to CDGCN. As shown in Table 2, we analyzed the gain from the CDGCN-based clustering method. First, we designed an initial speaker diarization system S6 with the Leiden clustering module only. The inputs of the system S6 are raw graphs, where every node pair has a weighted edge. Many node pairs are linked incorrectly, which causes the high Diarization Error Rate (DER) of the initial system. Secondly, we applied the KNN algorithm to ensure that only the edges between the pivot node and its top-K neighbors are well-connected. This operation ignored many wrong linkages and made the DER decrease rapidly. The GCN refines the linkages between nodes according to their neighbors by adding GCN refinement within a sub-graph context. After refinement, the DER is decreased from 17.04% to 15.38% on the Full EVAL dataset. We performed the Graph-OSD module to further improve the system's performance, and achieved a DER of 13.72% on the Full EVAL dataset. In order to evaluate the accuracy of second speaker labels produced by CDGCN, we used oracle OSD labels for the graph-OSD module.

Table 3: MSE of speaker number prediction with different clustering methods on EVAL dataset.

ID	Method	MSE
S1	AHC	3.80
S2	K-means	2.05
S3	NME-SC	2.05
S6	Raw-Leiden(ours)	4.45
S7	KNN-Leiden(ours)	2.38
S5	CDGCN(ours)	**1.67**

The results showed that the DER of the full EVAL dataset was improved from 13.72% to 8.94% significantly. This demonstrated that overlapped speech is a critical factor that limits system performance.

5.3. Speaker number prediction

In order to further evaluate the performance of clustering methods, we calculated the Mean Square Error (MSE) of speaker number prediction for the above clustering methods. As shown in Table 3, the CDGCN outperformed the traditional clustering methods on the speaker number prediction task. The inputs of the KNN-Leiden system are speaker graphs constructed by KNN algorithm. When compared the performance of KNN-Leiden and CDGCN algorithms, we can see that GCN model boosts the MSE from 2.38 to 1.67 on EVAL dataset. By optimizing the global quality function, CDGCN can find a more appropriate graph partition to predict the number of speakers.

6. CONCLUSIONS

This paper proposes a novel speaker clustering method based on the speaker topological graph for speaker diarization. We aim to give consideration to both local and global information when clustering. The proposed CDGCN-based clustering approach include graph generation, sub-graphs detection, and Graph-OSD. The local linkage between speech segments is inferred by a GCN model, while the Leiden community detection algorithm is applied to find the global partition of the speaker graph. To further improve the performance of our speaker diarization system, we also proposed a Graph-OSD component to handle overlapped speech for speaker diarization. Experimental results demonstrated that CDGCN based speaker diarization system outperformed conventional CSD systems in the DIHARD III corpus.

7. REFERENCES

[1] Christoph Boeddeker, Jens Heitkaemper, Joerg Schmalen-stroeer, Lukas Drude, Jahn Heymann, and Reinhold Haeb-Umbach, "Front-end processing for the chime-5 dinner party scenario," in *CHiME5 Workshop, Hyderabad, India*, 2018, vol. 1.

[2] Ivan Medennikov, Maxim Korenevsky, Tatiana Prisyach, Yuri Khokhlov, Mariya Korenevskaya, Ivan Sorokin, Tatiana Timofeeva, Anton Mitrofanov, Andrei Andrusenko, Ivan Podluzhny, et al., "The stc system for the chime-6 challenge," in *CHiME 2020 Workshop on Speech Processing in Everyday Environments*, 2020.

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

[5] Huazhong Ning, Ming Liu, Hao Tang, and Thomas S Huang, "A spectral clustering approach to speaker diarization," in *Ninth International Conference on Spoken Language Processing*, 2006.

[6] Gregory Sell, Alan McCree, and Daniel Garcia-Romero, "Priors for speaker counting and diarization with ahc.," in *Inter-Speech*, 2016, pp. 2194–2198.

[7] Xiong Xiao, Naoyuki Kanda, Zhuo Chen, Tianyan Zhou, Takuya Yoshioka, Sanyuan Chen, Yong Zhao, Gang Liu, Yu Wu, Jian Wu, et al., "Microsoft speaker diarization system for the voxceleb speaker recognition challenge 2020," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 5824–5828.

[8] Jie Wang, Yuji Liu, Binling Wang, Yiming Zhi, Song Li, Shipeng Xia, Jiayang Zhang, Feng Tong, Lin Li, and Qingyang Hong, "Spatial-aware speaker diarization for multi-channel multi-party meeting," in *Proc. Interspeech 2022*, 2022, pp. 1491–1495.

[9] Qingjian Lin, Yu Hou, and Ming Li, "Self-attentive similarity measurement strategies in speaker diarization.," in *INTERSPEECH*, 2020, pp. 284–288.

[10] Qingjian Lin, Ruiqing Yin, Ming Li, Hervé Bredin, and Claude Barras, "LSTM Based Similarity Measurement with Spectral Clustering for Speaker Diarization," in *Proc. Interspeech 2019*, 2019, pp. 366–370.

[11] Quan Wang, Carlton Downey, Li Wan, Philip Andrew Mansfield, and Ignacio Lopz Moreno, "Speaker diarization with lstm," in *2018 IEEE International conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2018, pp. 5239–5243.

[12] Jianbo Shi and Jitendra Malik, "Normalized cuts and image segmentation," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 22, no. 8, pp. 888–905, 2000.

[13] Max Welling and Thomas N Kipf, "Semi-supervised classification with graph convolutional networks," in *J. International Conference on Learning Representations (ICLR 2017)*, 2016.

[14] Fuchuan Tong, Siqi Zheng, Min Zhang, Yafeng Chen, Hongbin Suo, Qingyang Hong, and Lin Li, "Graph convolutional network based semi-supervised learning on multi-speaker meeting data," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 6622–6626.

[15] Jixuan Wang, Xiong Xiao, Jian Wu, Ranjani Ramamurthy, Frank Rudzicz, and Michael Brudno, "Speaker diarization with session-level speaker embedding refinement using graph neural networks," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 7109–7113.

[16] Neville Ryant, Prachi Singh, Venkat Krishnamohan, Rajat Varma, Kenneth Church, Christopher Cieri, Jun Du, Sriram Ganapathy, and Mark Liberman, "The third dihard diarization challenge," *arXiv preprint arXiv:2012.01477*, 2020.

[17] Zhongdao Wang, Liang Zheng, Yali Li, and Shengjin Wang, "Linkage based face clustering via graph convolution network," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 1117–1125.

[18] Siqi Zheng and Hongbin Suo, "Reformulating speaker diarization as community detection with emphasis on topological structure," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 8097–8101.

[19] V, A, Traag, L, Waltman, N, J, van, and Eck, "From louvain to leiden: guaranteeing well-connected communities.," *Scientific Reports*, 2019.

[20] Jörg Reichardt and Stefan Bornholdt, "Statistical mechanics of community detection," *Phys. Rev. E*, vol. 74, pp. 016110, Jul 2006.

[21] Hervé Bredin and Antoine Laurent, "End-to-end speaker segmentation for overlap-aware resegmentation," in *Proc. Interspeech*, 2021.

[22] Fuchuan Tong, Miao Zhao, Jianfeng Zhou, Hao Lu, Zheng Li, Lin Li, and Qingyang Hong, "ASV-Subtools: Open source toolkit for automatic speaker verification," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 6184–6188.

[23] Desh Raj, Leibny Paola Garcia-Perera, Zili Huang, Shinji Watanabe, Daniel Povey, Andreas Stolcke, and Sanjeev Khudanpur, "DOVER-Lap: A method for combining overlap-aware diarization outputs," in *2021 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2021, pp. 881–888.

[24] Federico Landini, Alicia Lozano-Diez, Lukáš Burget, Mireia Diez, Anna Silnova, K Zmolıková, O Glembek, P Matejka, T Stafylakis, and N Brümmer, "But system description for the third dihard speech diarization challenge," in *Proc. 3rd DIHARD Speech Diarization Challenge Workshop*, 2021.

[25] Shota Horiguchi, Nelson Yalta, Paola Garcia, Yuki Takashima, Yawen Xue, Desh Raj, Zili Huang, Yusuke Fujita, Shinji Watanabe, and Sanjeev Khudanpur, "The hitachi-jhu dihard iii system: Competitive end-to-end neural diarization and x-vector clustering systems combined by dover-lap," *arXiv preprint arXiv:2102.01363*, 2021.

[26] Tae Jin Park, Kyu J Han, Manoj Kumar, and Shrikanth Narayanan, "Auto-tuning spectral clustering for speaker diarization using normalized maximum eigengap," *IEEE Signal Processing Letters*, vol. 27, pp. 381–385, 2019.