Graph Clustering approaches for Speaker Diarization of Conversational Speech

PhD Thesis Colloquium
21st July 2023

Prachi Singh
PhD Student,
Learning and Extraction of Acoustic Patterns (LEAP) Lab,
Electrical Engineering, Indian Institute of Science, Bangalore.

Advisor: Dr. Sriram Ganapathy
Outline

• Introduction
  • Motivation
  • Methodology
  • Contributions

• Background study
  • Related work
  • Performance metrics
  • Datasets

• Proposed Graph Clustering approaches

• Conclusion and Future Directions
Introduction
Motivation

What is a conversational speech?

Conversational audio contains multiple speakers engaged in a conversation.

Modelling of such audio requires understanding speakers’ characteristics and content.
Motivation

Transcribing audio into text using speaker information generates much meaningful text.

The task of finding “who spoke when” is called Speaker Diarization.

Transcribing meeting

Call center interactions Analysis
Definition

Speaker diarization is the task of partitioning an input audio recording into segments based on speakers and assign relative speaker labels.
Methodology

Sell et al., Diarization is hard: some experiences and lessons learned for the JHU team in the inaugural DIHARD challenge, 2018.
Contributions outline

Input Audio

- Speech Activity Detection
- Segmentation
- Feature extraction

Post processing - Temporal smoothing

Clustering

Speaker embedding extraction

Focus of this thesis

Diarization output

Speaker1 Speaker2 Speaker3
Contributions outline

• Clustering is a crucial step in speaker diarization as it enables
  • Accurate speaker segmentation
  • Turn-taking detection
  • Speaker model creation
  • Speaker adaptation, and evaluation
• Improving speaker embeddings can help improve clustering
Contributions outline

Application of graph models to temporal segmentation of speech is the first of its kind.

- Novel hierarchical graph clustering
- Self-supervised metric learning to generate similarity for clustering
- Supervised hierarchical graph clustering
Background study
Related work

Unsupervised Clustering approaches

Forming groups based on hidden patterns in the unlabeled data

• Hierarchical clustering

• Graph Clustering
Unsupervised Clustering approaches

Hierarchical clustering

• Clusters are visually represented in a hierarchical tree called a dendrogram.

Example: Agglomerative Hierarchical clustering (AHC)
Clustering approaches

Graph

- A graph G can be well described by the set of **vertices** V and **edges** E it contains. G=(V,E)
- The **vertices** are often called **nodes**.

- **Adjacency matrix** (A) captures connections between nodes,
  - \( A_{ij} = 1, \text{if Node } i \text{ is connected to } j \text{ by an edge} \)
  - \( A_{ij} = 0, \text{if Node } i \text{ and } j \text{ are not connected} \)
- A with real weights to the edges is called as weighted adjacency matrix.

Graph clustering

Clustering the nodes such that **many edges** are present **within each cluster** and **fewer edges between the clusters**.

Example: Spectral Clustering (SC)
Can we combine the two?

• Why not!

- Hierarchical Graph Clustering
Related work

• Speaker embeddings/representations
  • i-vector\textsuperscript{1} – statistical model
  • d-vector\textsuperscript{2} – Deep Neural Network
  • x-vector\textsuperscript{3} – Time delay Neural Network (widely used)

• Similarity measure
  • Cosine\textsuperscript{4}
  • PLDA\textsuperscript{5} (widely used)
Related work

End to end neural diarization (EEND)\(^1\)

- Transformer is used to perform speaker activity detection
- Takes input as F-dimensional audio features and generate C speaker labels

Cons:
- Requires huge amount of labelled data for training.
- Difficult to generalize for higher number of speakers.
- Cannot handle long duration recording at a time.

\(^1\)Horiguchi et. al., “End-to-End Speaker Diarization for an Unknown Number of Speakers with Encoder-Decoder Based Attractors
Performance metric

Optimal mapping: \( \text{argmax}(A \cap 1, A \cap 2), \text{argmax}(B \cap 1, B \cap 2) \)
Performance metric

• **Diarization error rate (DER)** is the standard metric for evaluating and comparing speaker diarization systems.

• It is defined as follows:

\[
DER = \frac{\text{false alarm} + \text{miss detection} + \text{speaker confusion}}{\text{total speakers duration}}
\]

• *false alarm* - duration of non-speech predicted as speech

• *miss detection* - duration of speech of a speaker predicted as non-speech

• *speaker confusion* – duration of a reference speaker predicted as another speaker in system output after optimal mapping

• *total speakers duration* – total duration of all the speakers present
# Test Datasets

## Narrowband (sampling rate: 8kHz)

<table>
<thead>
<tr>
<th>CALLHOME (CH) [1]</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Multi-lingual telephone data</td>
</tr>
<tr>
<td>- Recordings - 500, CH1 – 250, CH2 - 250,</td>
</tr>
<tr>
<td>- 2-5 mins</td>
</tr>
<tr>
<td>- 2-7 speakers</td>
</tr>
</tbody>
</table>

## Wideband (sampling rate: 16kHz)

<table>
<thead>
<tr>
<th>AMI [2]</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Augmented Multi-party Interactions</td>
</tr>
<tr>
<td>- Recorded at four different sites (Edinburgh, Idiap, TNO, Brno)</td>
</tr>
<tr>
<td>- Recordings - Dev set: 18, Eval set: 16</td>
</tr>
<tr>
<td>- 20-60mins</td>
</tr>
<tr>
<td>- 3-4 speakers</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DIHARD III [3]</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Speech diarization challenge data</td>
</tr>
<tr>
<td>- 9-11 domains e.g, audiobooks, telephone recording, meetings, web videos.</td>
</tr>
<tr>
<td>- Recordings – Dev set:254, Eval set:259</td>
</tr>
<tr>
<td>- 0.5-10 mins</td>
</tr>
<tr>
<td>- 1-10 speakers</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Voxconverse [4]</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Voxconverse challenge data</td>
</tr>
<tr>
<td>- Conversational dataset extracted from YouTube videos.</td>
</tr>
<tr>
<td>- dev set: 216 and eval set:232</td>
</tr>
<tr>
<td>- 22s - 20mins.</td>
</tr>
<tr>
<td>- 1-21 speakers.</td>
</tr>
</tbody>
</table>

---

[2] McCowan et al., The AMI meeting corpus, 2005
[3] Ryant et al., The Third DIHARD Diarization Challenge, 2020
Proposed Approach 1

- Introduced **self-supervised learning using DNN**.
- Introduced **hierarchical graph clustering**.

**Self-Supervised Clustering**

- Supervised clustering using graph neural networks

**Self-supervised Clustering with Metric Learning**
Motivation

• The clustering approaches extract short-segment speaker embeddings from a pre-trained network (x-vectors) and perform unsupervised clustering.

• Each stage (embedding extraction and clustering) is optimized independently.

• The test set will contain unseen domains and speakers.
Motivation

- **Succinct speaker representations** are beneficial for clustering while clustering results can provide self-supervisory targets for representation learning¹.

- Creating a **feedback loop** from the output of the clustering algorithm to the input can help improve the representations used for clustering.

- This is referred to as **self-supervised clustering (SSC)**.

- The data itself provides supervision labels for model learning

¹Yang et. al., “Joint unsupervised learning of deep representations and image clusters,” in CVPR, 2016
Self-Supervised Clustering alternates between merging the clusters for a fixed embedding representation and learning the representations using the given cluster labels, till we reach the required number of clusters/speakers.
Variables:
\[ X = \{x_1, \ldots, x_{N_r}\} \in \mathbb{R}^D: \text{X-vectors sequence of recording } r \]
\[ Y = \{y_1, \ldots, y_{N_r}\} \in \mathbb{R}^d: \text{lower dimensional representations} \]
\[ z = \{z_1, \ldots, z_{N_r}\} \in \mathbb{R}: \text{segment labels} \]
\[ \theta: \text{DNN parameters} \]
\( (Y^q, z^q, \theta^q): \text{refer to variables at iteration } q \)
\[ N^q: \text{Number of clusters at iteration } q \]
\[ N^*: \text{target number of clusters} \]

For DNN training at iteration \( q \), use clustering results from \( q-1 \) to sample positive and negative pairs of triplets.

DNN training—Triplet loss

• For each cluster $C_i^q$, pick two elements as anchor and positive {$y_i, y_j$}.

• For negative pair, element {$y_l$} is selected randomly from any other cluster.

• Triplet loss:

$$\theta^q = \arg\max_\theta \sum_{i,j,l} [s(i,j) - \alpha(s(i,l) + s(j,l))]$$

$s(i,j)$ — similarity score; $0 < \alpha \leq 1$ : weighting factor
Agglomerative clustering

AHC

Merging Criterion:

In an AHC algorithm, the merging criterion for merging two clusters $C_a^q$ and $C_b^q$ where $q$ is the iteration index is given as

$$\{ C_a^q, C_b^q \} = \arg \max_{C_i, C_j \in C, i \neq j} A(C_i, C_j)$$

(where, $A$ denote the affinity measure between two clusters.)
Agglomerative clustering

Path integral clustering (PIC)

Graph-structural based agglomerative clustering algorithm where graph encodes the structure of the embedding space.

1. Measures the affinity of clusters based on the neighborhood graph hence is more robust to noisy distances.

2. Uses robust graph structural merging strategy for noisy links.

3. It does not assume anything on the underlying data distributions and only need the pairwise similarities of samples.
Path Integral Clustering (PIC)

- Given a set of vectors $X = \{x_1, x_2, \ldots, x_n\}$, it involves creation of directed graph $G = (V,E)$

- Weighted Graph Adjacency matrix ($W$) given as,

\[
W_{ij} = S(i,j) \text{ if } x_j \in N^K_i
\]

\[
= 0 \text{ otherwise}
\]

where, $S(i,j)$ is the pairwise similarity between $x_i$ and $x_j$, $N^K_i$ is the set of K nearest neighbour of $x_i$
\[ \mathcal{A}_{c_a,c_b} = (S_{c_a|c_a\cup c_b} - S_{c_a}) + (S_{c_b|c_a\cup c_b} - S_{c_b}). \]
## Baselines

<table>
<thead>
<tr>
<th>Step</th>
<th>Parameter</th>
<th>CH</th>
<th>AMI</th>
</tr>
</thead>
<tbody>
<tr>
<td>-</td>
<td>Sampling rate</td>
<td>8kHz</td>
<td>16kHz</td>
</tr>
<tr>
<td>Segmentation</td>
<td>Window size</td>
<td>1.5s, 0.75s shift</td>
<td>1.5s, 0.75s shift</td>
</tr>
<tr>
<td>Embedding extraction (x-vector) extraction</td>
<td>Architecture</td>
<td>7-layers TDNN</td>
<td>7-layers TDNN</td>
</tr>
<tr>
<td></td>
<td>Train set</td>
<td>SWBD, SRE</td>
<td>Voxceleb 1,2</td>
</tr>
<tr>
<td></td>
<td>Train #speakers</td>
<td>4,285</td>
<td>7,323</td>
</tr>
<tr>
<td></td>
<td>Input features</td>
<td>23D MFCCs</td>
<td>30D MFCCs</td>
</tr>
<tr>
<td></td>
<td>x-vector</td>
<td>128</td>
<td>512</td>
</tr>
<tr>
<td></td>
<td>dimension</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Similarity score</td>
<td>type</td>
<td>PLDA</td>
<td>PLDA</td>
</tr>
<tr>
<td>Clustering</td>
<td>type</td>
<td>AHC</td>
<td>AHC</td>
</tr>
</tbody>
</table>
Implementation details

<table>
<thead>
<tr>
<th>config</th>
<th>CH</th>
<th>AMI</th>
</tr>
</thead>
<tbody>
<tr>
<td>x-vectors/recording</td>
<td>50-700</td>
<td>1000-4000</td>
</tr>
<tr>
<td>2-layer DNN</td>
<td>128x10</td>
<td>512X30</td>
</tr>
<tr>
<td>Learning rate</td>
<td>0.001</td>
<td>0.001</td>
</tr>
<tr>
<td>Annealing</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Batch</td>
<td>Full</td>
<td>Mini-batch</td>
</tr>
<tr>
<td>epochs</td>
<td>5-10</td>
<td>5-10</td>
</tr>
</tbody>
</table>
Initialization

• Weight initialization and training are file specific

• Uses processing steps from baseline system for PLDA scoring

• First layer is initialized using global PCA computed using held out set followed by length norm.

• Second layer is initialized using file-level PCA

• Affinity measure : Cosine similarity
CH Results

- Performance metric: Diarization Error Rate (DER) (%)

- Considering only non-overlapping speech regions with tolerance collar (0.25s).

<table>
<thead>
<tr>
<th>System</th>
<th>Known N*</th>
<th>Unknown N*</th>
</tr>
</thead>
<tbody>
<tr>
<td>x-vec + cosine + AHC</td>
<td>8.9</td>
<td>10.0</td>
</tr>
<tr>
<td>x-vec + cosine + SC</td>
<td>9.4</td>
<td>11.9</td>
</tr>
<tr>
<td>x-vec + PLDA + AHC</td>
<td>7.0</td>
<td>8.0</td>
</tr>
<tr>
<td>x-vec + cosine + PIC</td>
<td>7.7</td>
<td>9.3</td>
</tr>
<tr>
<td>SSC-AHC</td>
<td>6.4</td>
<td>8.3</td>
</tr>
<tr>
<td>SSC-PIC</td>
<td>6.4</td>
<td>7.5</td>
</tr>
<tr>
<td>+ Temp. cont.</td>
<td><strong>6.3</strong></td>
<td><strong>7.0</strong></td>
</tr>
</tbody>
</table>
## AMI Results

<table>
<thead>
<tr>
<th>System</th>
<th>Known N*</th>
<th>Unknown N*</th>
</tr>
</thead>
<tbody>
<tr>
<td>x-vec + cosine + AHC</td>
<td>34.6</td>
<td>30.2</td>
</tr>
<tr>
<td>x-vec + cosine + SC</td>
<td>30.2</td>
<td>25.5</td>
</tr>
<tr>
<td>x-vec + PLDA + AHC (Baseline)</td>
<td>15.7</td>
<td>16.0</td>
</tr>
<tr>
<td>SSC-PLDA-AHC</td>
<td>9.4</td>
<td>11.1</td>
</tr>
<tr>
<td>x-vec + PLDA + PIC</td>
<td>9.4</td>
<td>9.3</td>
</tr>
<tr>
<td>x-vec + cosine + PIC</td>
<td>8.9</td>
<td>7.3</td>
</tr>
<tr>
<td>SSC-PIC + Temp. cont.</td>
<td>7.3</td>
<td>7.2</td>
</tr>
</tbody>
</table>

AMI Visualization

Baseline Embeddings

SSC Embeddings

F-score: 4.51

F-score: 8.72

(a) t-SNE based visualization of embeddings extracted on 1.5s audio segments from the meeting dataset.

# AMI Results

## DER comparison with other published works

<table>
<thead>
<tr>
<th>System</th>
<th>Known N*</th>
<th></th>
<th>Unknown N*</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Semi-sup learning(^1)</td>
<td>-</td>
<td>-</td>
<td>17.5</td>
<td>22.0</td>
</tr>
<tr>
<td>Incremental(^2) learning</td>
<td>-</td>
<td>15.6</td>
<td>-</td>
<td>20.0</td>
</tr>
<tr>
<td><strong>GAN clustering</strong>(^3)</td>
<td>10.2</td>
<td>10.1</td>
<td>11.0</td>
<td>11.3</td>
</tr>
<tr>
<td>2D self-attention(^4)</td>
<td>-</td>
<td>-</td>
<td>12.2</td>
<td>13.0</td>
</tr>
<tr>
<td>Baseline</td>
<td>14.4</td>
<td>16.5</td>
<td>12.9</td>
<td>13.6</td>
</tr>
<tr>
<td><strong>SSC-PIC</strong></td>
<td>4.6</td>
<td>6.5</td>
<td>5.2</td>
<td>5.4</td>
</tr>
</tbody>
</table>

---

1. Pal et al., A study of semi-supervised speaker diarization system using GAN mixture mode, 2019
2. Dawalatabad et al., Incremental Transfer Learning in Two-pass Information Bottleneck Based Speaker Diarization System for Meetings, 2019
3. Pal et al., Speaker diarization using latent space clustering in generative adversarial network, 2020
4. Sun et al., Speaker diarisation using (2D) self-attentive combination of embeddings, 2019
Summary

• Proposed **self-supervised clustering algorithm using DNN** which iteratively updates representation learning and clustering.

• Introduced **path integral clustering – hierarchical graph clustering for first time for diarization**.

• Encourages separation between representations of different speakers.

• Improvements on AMI and CALLHOME dataset.
Proposed Approach 2

Graph Clustering

Self-Supervised Clustering

Supervised clustering using graph neural networks

Self-supervised Clustering with Metric Learning
Motivation

• **SSC uses cosine similarity** to perform clustering.

• **Prior work on clustering performs better with PLDA score than cosine.**

• **PLDA**\(^1\) is a parametric model which is trained using Expectation Maximization (EM).

• **Can we train the SSC with learnable scoring/metric function?**
  • Yes. SelfSup-PLDA-PIC.
SelfSup-PLDA-PIC

• Self-supervised metric learning with graph-based clustering algorithm (SelfSup-PLDA-PIC) jointly performs representation learning and metric learning using the initial clustering results.

• Propose a neural version of PLDA to incorporate deep learning of the PLDA model parameters.

Block diagram: SelfSup-PLDA-PIC
Metric Learning using PLDA model

• Probabilistic Linear Discriminant Analysis (PLDA)\(^1\) is a supervised generative model trained to learn distributions of different speakers.

• It can be used to find pairwise similarity score between embeddings from unseen speakers as follows

\[
s(i, j) = \log \left[ \frac{p(u_i, u_j | H_s)}{p(u_i | H_d)p(u_j | H_d)} \right]
\]

Same-speaker hypothesis

different-speakers hypothesis
Metric Learning using PLDA model

- Replacing PLDA model with a learnable parametric model with parameter $\Psi$

\[ f(u_i, u_j, \Psi) \]
AMI Results

**AMI DER (%) Results** – Ignoring overlaps and with collar 0.25s

<table>
<thead>
<tr>
<th>System</th>
<th>Known N*</th>
<th>Unknown N*</th>
</tr>
</thead>
<tbody>
<tr>
<td>x-vec + PLDA + AHC</td>
<td>15.9</td>
<td>12.2</td>
</tr>
<tr>
<td>x-vec + PLDA + PIC</td>
<td>5.1</td>
<td>10.2</td>
</tr>
<tr>
<td>SSC-Cosine-PIC</td>
<td>5.3</td>
<td>6.2</td>
</tr>
<tr>
<td>SelfSup-PLDA-AHC</td>
<td>7.9</td>
<td>7.3</td>
</tr>
<tr>
<td>SelfSup-PLDA-PIC$^1$ + Temporal continuity</td>
<td>4.2</td>
<td>6.2</td>
</tr>
<tr>
<td>SelfSup-PLDA-PIC + VBx$^2$</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

$^2$Diez et al., Bayesian HMM based x-vector clustering for speaker diarization, 2019
Similarity score matrices comparison for 4-speaker recording from AMI development set
DIHARD Results

Average DER (%) on the DIHARD dataset considering overlapping regions with no tolerance collar.

For recordings with ≤ 7 speakers and > 7 speakers.

<table>
<thead>
<tr>
<th>System</th>
<th>≤ 7 speakers</th>
<th>&gt; 7 speakers</th>
</tr>
</thead>
<tbody>
<tr>
<td>X-vec + PLDA + AHC</td>
<td>18.0</td>
<td>19.3</td>
</tr>
<tr>
<td>X-vec + PLDA + PIC</td>
<td>17.7</td>
<td>17.8</td>
</tr>
<tr>
<td>SelfSup-PLDA-PIC</td>
<td>17.0</td>
<td>17.2</td>
</tr>
</tbody>
</table>
Summary

• Proposed **self-supervised metric learning** approach using **PLDA**.

• Increases inter-speaker distance and decreases intra-speaker distance.

• Performance degrades as number of speakers increases as initial clustering becomes unreliable.
Proposed Approach 3

Graph Clustering

Self-Supervised Clustering

Supervised clustering using graph neural networks
First of its kind attempt to perform supervised hierarchical clustering for diarization

Self-supervised Clustering with Metric Learning
Motivation

• Self-supervised clustering is less reliable when recording contains higher number of speakers (>7).

• The end goal is to minimize the clustering errors to improve performance.

• Can we train a model with the clustering objective?
Supervised HierArchical GRaph Clustering (SHARC)

- Performs **supervised clustering** using **Graph Neural Networks (GNN)**.
- Represents the speaker embeddings using graph.
- Clustering loss is used to update edges of the graph.
- Generates node labels based on clustering performed on updated edges at each level of hierarchy.
SHARC components

• Graph Generation
• GNN scoring
• Clustering
• Aggregation
Graph generation

Test recording

ETDNN Model

\(X_t\)

Similarity matrix (\(S_t\))

k-nn graph

k=2 nearest neighbors

X1  X2  \ldots  Xn

X1

X2

\ldots

Xn
GNN scoring

• GNN scoring function $\Psi$ - a learnable GNN module designed for supervised clustering.

• Output: edge prediction probability $p_{ij}$ between node $i$ and $j$.

• $N_i^k$ - $k$-nearest neighbors of node $v_i$,

$$\hat{e}_{ij} = 2p_{ij} - 1 \in [-1, 1] \forall j \in N_i^k$$

• Density of node $i$:

• Ground truth:

$$d_i = \frac{1}{k} \sum_{j \in N_i^k} \hat{e}_{ij} S_r(i, j)$$

• Predicted:

$$\hat{d}_i = \frac{1}{k} \sum_{j \in N_i^k} \hat{e}_{ij} S_r(i, j)$$
Clustering

• At each level of hierarchy m, it creates a candidate edge set $\varepsilon(i)$

$$\varepsilon(i) = \{ j | (v_i, v_j) \in E_m, \hat{d}_i \leq \hat{d}_j \text{ and } p_{ij} \geq p_r \}$$

• For any i, if $\varepsilon(i)$ is not empty, we pick $j = \arg\max_{j \in \varepsilon(i)} \hat{e}_{ij}$ and add $(v_i, v_j)$ to $E'_m$

• A set of connected components $C'_m$, forms clusters for the next level (m + 1).
Feature Aggregation

The aggregation function $Ψ$ - obtain node representations for next level.

$^1$Prachi Singh, Amrit Kaul, Sriram Ganapathy, “Supervised Hierarchical Clustering Using Graph Neural Networks For Speaker Diarization”, accepted in ICASSP 2023
Block diagram: SHARC Inference

SHARC Components
1. Graph Generation
2. GNN scoring
3. Clustering
4. Aggregation
Training loss

- Loss: \( L = L_{\text{conn}} + L_{\text{den}} \)
  - \( L_{\text{conn}} = \frac{1}{|E|} \sum_{i,j \in E} q_{ij} \log p_{ij} + (1 - q_{ij}) \log (1 - p_{ij}) \)
    \( q_{ij} \) - Ground truth edge labels, \( p_{ij} \) - predicted edge labels
  
  \( L_{\text{den}} = \frac{1}{|V|} \sum_{i=1}^{|V|} \| d_i - \hat{d}_i \|^2_2 \) \( \forall i \in \{1, ..., |V| \} \), where \( |V| \) is the cardinality of \( V \)
Experiments

Datasets

AMI: Meeting dataset

Voxconverse: Youtube videos
Results

- Performance: DER (%) (lower the better)

53% relative improvement over best baseline

Results

• Performance: DER (%) (lower the better)

Voxconverse dataset

- E-SHARC + Refinement
- E-SHARC
- Proposed (SHARC)
- Baseline 2 Spec. clustering
- Baseline 1 AHC

41% relative improvement over best baseline
## Cluster Purity and Coverage

**Purity:** The percentage of segments from predicted speaker belong to one speaker in ground truth.

**Coverage:** The percentage of segments from ground truth speaker is covered by predicted speaker.

### Voxconverse

<table>
<thead>
<tr>
<th>Method</th>
<th>Purity</th>
<th>Coverage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline with AHC</td>
<td>93.5</td>
<td>89.5</td>
</tr>
<tr>
<td>Baseline with SC</td>
<td>92.0</td>
<td>92.3</td>
</tr>
<tr>
<td>SHARC</td>
<td>93.0</td>
<td>92.4</td>
</tr>
<tr>
<td>E-SHARC</td>
<td>93.0</td>
<td>92.9</td>
</tr>
</tbody>
</table>
## Results

Results with pyannote overlap detection

<table>
<thead>
<tr>
<th>AMI</th>
<th>Eval DER (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>AHC + overlap</td>
<td>26.30</td>
</tr>
<tr>
<td>SC + overlap</td>
<td>18.10</td>
</tr>
<tr>
<td>SHARC + overlap</td>
<td>19.32</td>
</tr>
<tr>
<td>E-SHARC + overlap</td>
<td>17.95</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Voxconverse</th>
<th>Eval DER (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>AHC + overlap</td>
<td>11.66</td>
</tr>
<tr>
<td>SC + overlap</td>
<td>10.73</td>
</tr>
<tr>
<td>SHARC + overlap</td>
<td>10.89</td>
</tr>
<tr>
<td>E-SHARC + overlap</td>
<td>10.44</td>
</tr>
</tbody>
</table>

Bredin et al., pyannote.audio: neural building blocks for speaker diarization, 2020

85
Summary

• Introduced **supervised hierarchical clustering** for speaker diarization for the first time.

• Designed an **end-to-end approach** to perform speaker diarization using **Graph Neural Networks**.

• Achieved state-of-the-art performance on two benchmark datasets.
Conclusion and Future Directions
Real world Example
Results

- Groundtruth
- Baseline (DER: 35.64%)
- Self-supervised Approach (DER: 21.22%)
- Supervised Approach (DER: 15.10%)
Concluding remarks

<table>
<thead>
<tr>
<th>Proposed Approaches</th>
<th>Novelties</th>
<th>Limitations</th>
</tr>
</thead>
</table>
| SSC                 | • Introduced self-supervised clustering using DNN  
                      • Introduced PIC graph clustering for the first time to improve diarization. | • Similarity scoring is not learnable (cosine)                               |
| SelfSup-PLDA-PIC    | • Introduced self-supervised metric learning    | • Performance depends on initial clustering                                  
                                                                                           • Degrades with higher number of speakers                                          |
| SHARC               | • First time performed supervised hierarchical clustering for diarization | • Increased training time                                                   
                                                                                           • Require domain specific training                                                
                                                                                           • Not purely end-to-end                                                             
                                                                                           • Overlap detection can be added                                                   |
Future Directions

Multilingual conversation Diarization

Use Multi-edge graph to perform multi-task learning

Source: https://displace2023.github.io/
Future Directions

Target speaker identification in conversational speech

Enrollment
(Example of how Ramesh sounds like)

Test recording

Is Ramesh present in test?
Yes

Possible direction

Graph clustering approaches
Yes

- Need to handle channel mismatch
- Avoid clustering within target speaker recording
Publications based on the thesis

• Peer-reviewed Journals:
  • P. Singh and S. Ganapathy, “Speaker Diarization with Graph Based Supervised Hierarchical Clustering” (under preparation).

• Peer-reviewed Conferences: