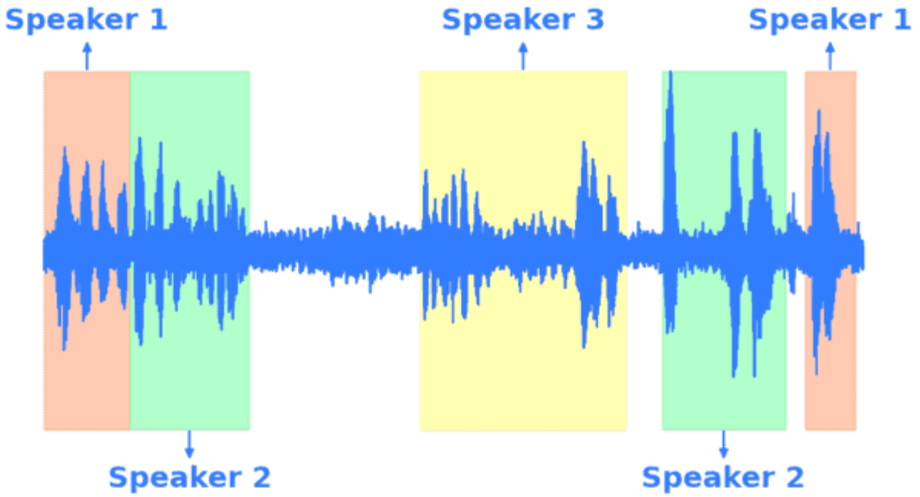
EE698r Project

# Speaker Diarization




Made by:-
Krishiv Geriani(220545)
Vishap Raj(221208)

## PROBLEM STATEMENT

The goal of this diarization project is to develop a system that can accurately segment and label the speakers in an audio recording, identifying "who spoke when." This is commonly referred to as speaker diarization. The system should be able to distinguish multiple speakers in a conversation, track each speaker's activity over time, and output the timing and identities of each speaker.

## EVALUATION METRICS

To evaluate the performance of the diarization system, we will use the following metrics:

1. **Diarization Error Rate:**

$$DER = \frac{(FalseAlarms + Misses + Confusion)}{TotalTime}$$

The primary metric for diarization systems. A lower DER indicates better performance.

2. **Precision and Recall**

Precision measures the fraction of time correctly attributed to a specific speaker, and Recall measures the fraction of time a specific speaker was correctly identified.

3. **F1 Score**

The harmonic mean of Precision and Recall, giving a balanced measure of the system’s performance.

# DATASETS AVAILABLE

The performance of speaker diarization models heavily depends on the quality and diversity of the datasets used.

## AMI Meeting Corpus

https://groups.inf.ed.ac.uk/ami/corpus/

The AMI Meeting Corpus is a multi-modal data set consisting of 100 hours of meeting recordings.

## CALLHOME American English Speech

https://paperswithcode.com/dataset/callhome-american-english-speech

The CALLHOME English Corpus is a collection of unscripted telephone conversations between native speakers of English.The corpus contains 120 telephone conversations, each lasting up to 30 minutes.

## Movie Sound Clips

https://www.moviesoundclips.net/

# REFERENCES

https://github.com/pyannote/pyannote-audio

pyannote.audio is an open-source toolkit written in Python for speaker diarization. Based on PyTorch machine learning framework, it comes with state-of-the-art pretrained models and pipelines, that can be further finetuned to your own data for even better performance.

https://github.com/hitachi-speech/EEND

EEND (End-to-End Neural Diarization) is a neural-network-based speaker diarization method.

## Pre-Trained Models

**1.NVIDIA NeMo Diarization Models:**

https://docs.nvidia.com/nemo-framework/user-guide/latest/nemotoolkit/asr/speaker_diarization/results.htmla#end-to-end-speaker-diarization-models

**2.Pyannote.audio: segmentation Models**

# Methods

After evaluating the techniques , the most effective combination for speaker diarization can be :

1.CNN-based embeddings- For accurate speaker representations.

2.HMM(Hidden Markov Model)-For temporal modeling of speaker transitions.

## Advantages-

Better feature extraction → CNNs capture more detailed and robust speaker characteristics compared to MFCCs or basic embeddings.

Improved clustering → With richer embeddings, clustering (DBSCAN or HMM) becomes more accurate.

***HMM for Temporal Modeling-***

Accurate speaker transitions → HMM models the temporal dependencies between audio segments, making the diarization smoother.

Sequential consistency → Unlike DBSCAN, HMM understands the time continuity of a speaker, preventing rapid misclassifications in consecutive segments.

# GNATT CHART

| Task | DONE | Wk13 | Wk14 | Wk15 | Wk16 |
|---|---|---|---|---|---|
| **Data Preparation** | Collected datasets and preprocessed audio. | Extracted MFCC features. | Generated CNN embeddings. | | |
| **Embedding Techniques** | CNN embeddings with pretrained model. | Explore advanced embedding techniques. | Improve embedding accuracy. | | |
| **Clustering Techniques** | | Implement HMM clustering. | Optimize HMM parameters. | Explore better clustering methods. | Validate clustering accuracy. |
| **Testing and Evaluation** | | Prepare evaluation metrics. | Compare techniques. | Fine-tune performance. | Finalize and document results. |

# Approach for Speaker Diarization

**1. Preprocessing:**

Audio Loading: Load the audio file and convert it to mono.

Segmentation: Divide the audio into 1-3 second segments with time frames.

**2. Feature Extraction:**

MFCC Features: Extract Mel-Frequency Cepstral Coefficients (MFCC) to represent the spectral properties of each segment.

**3. Speaker Embeddings Using CNN:**

Use ECAPA-TDNN (CNN-based) pre-trained model to generate 192-dimensional speaker embeddings.

The embeddings capture distinct speaker characteristics.

**4. Clustering with HMM:**

Perform Hidden Markov Model (HMM) clustering to group segments by speaker.

Use BIC (Bayesian Information Criterion) to dynamically estimate the optimal number of speakers.

Cluster embeddings into different speaker groups.

**5. Speaker Labeling:**

Assign labels to each segment (e.g., Speaker 1, Speaker 2, Speaker 1).

Display speaker labels with corresponding time frames.

# THANK YOU