Name: Chowdam Venkata Thirumala Kumar
Program: MS by Research in ECE
Problem Statement: Speaker diarization

1. Literature

Speaker diarization is the task of identifying and separating different speakers in an audio recording. Speaker diarization has been addressed in two approaches,

1. **Modularized approach**: Speaker diarization consisting of multiple independent modules in sequential manner such as Voice Activity Detection (VAD), Segmentation, Embedding, Clustering, Post-processing. All modules are independently trained and used together for diarization. Initially VAD is performed to identify speech/non-speech segments followed by segmentation of speech segments. These segments will be passed through a speaker embedding model to extract embeddings. A clustering algorithm is used to cluster speech segments belonging to the same speakers. Optionally post-processing techniques can be employed to further refine the clustering results.
2. **End-to-End approach:** In this approach a single Neural Network model will be trained to perform all above mentioned tasks or jointly trained to optimise speaker diarization objective.

Speaker diarization tak is mainly evaluated through Diarization Error rate (DER). This metric is calculated by dividing the sum of False Alarms (Non-speech detected as speech) duration, Missed Detection (Speech detected as non-speech) duration and Speaker Confusion by total duration. Another metric Word-level Diarization Error Rate (WDER) is also considered when assessing both Automatic Speech Recognition and Speaker Diarization together or along with DER.

(Due to the lack of time, I couldn't explore the literature in detail and the above description is based on my own understanding.)

2. Dataset

Dataset consists of 57 audio recordings of doctor and patient conversations. It contains separate audios for doctor and patients speaking with a textgrid file containing time aligned transcripts individually. Combining the individual doctor and patient audio recordings as well as textgrid files will result in mixed audio and diarisation information respectively about who spoke when. This information will be the ground truth time stamps for speaker diarization. Data preprocessing includes (1) Mixing of doctor and patient recordings, (2) Generating combined time stamps information in rttm format (which is commonly used for speaker diarization).

3. Explored methods

Due to the lack of time, I couldn’t try finetuning the existing models using the given dataset. Also couldn’t explore the following methods in detail about their working flow.

1. **Pyannote**
    a. Pyannote v2.1
    b. Pyannote v3.1
2. **Reverb Diarization**

    Reverb diarization models are built on top of Pyannote framework

    a. Rev v1
    b. Rev v2
3. **Nemo Diarization**

    Nemo diarizer follows the modularized approach as mentioned in the Literature section. It does the clustering at multiple scales of segmentation for more accurate results, because speaker embeddings at longer duration segments will better represent whereas smaller duration scales will be better for better precision. Results at different scales will be combined. Clustering diarization consists of modules till clustering in the modularized approach as mentioned above. Neural diarization consists of the same modules till clustering along with a neural network module to refine clustering results as a post-processing module.

    a. Clustering diarization
    b. Neural diarization

4. Results and analysis

S No	Model	DER (in %)
1	Pyannote v2.1	22.83
2	Pyannote v3.1	26.23
3	Rev v1	20.49
4	Rev v2	21.90
5	Nemo Clustering Diarizer	23.12
6	Nemo Neural DIarizer	23.93

Above table shows the performance of different types of speaker diarization approaches/models available publicly in terms of diarization error rate (DER). Entire dataset of PriMock57 is used for calculating DER as I didn’t perform any finetuning. It can be observed that the Rev v1 model showed best DER than all other models with Rev v2 model being second best. It can be observed that Nemo Neural Diarizer couldn’t improve the clustering diarization performance. Since I haven’t gone through the in depth working flow of Pyannote models, I couldn’t compare the architectural complexities with respect to performance.