Mini Project Report

on

Diarizing Patient-Doctor Conversations

Submitted by

A. Tulasi narayana rao	B. Sreeshanth	D. Sathvik Reddy	M. Navadeep
22BDS004	22BDS016	22BDS020	22BDS040

Under the guidance of

Dr. Krishnendu Ghosh

Designation

DEPARTMENT OF DATA SCIENCE AND ARTIFICIAL INTELLIGENCE

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY DHARWAD

14/11/2025

Certificate

This is to certify that the project entitled **Diarizing Patient-Doctor Conversations** is a bonafide record of the Mini Project coursework presented by the students whose names are given below during the academic year **2025**, in partial fulfilment of the requirements for the degree of Bachelor of Technology in Data Science and Artificial Intelligence.

Roll No	**Name of Student**
22BDS004	A. Tulasi narayana rao
22BDS016	B. Sreeshanth
22BDS020	D. Sathvik Reddy
22BDS040	M. Navadeep

Dr. Krishnendu Ghosh
(Project Supervisor)

Contents

List of Figures

1 Introduction

Communication between doctors and patients forms the backbone of healthcare. During consultations, patients describe symptoms, medical history, lifestyle habits, and concerns, while doctors provide diagnosis explanations, treatment recommendations, medication schedules, and follow-up guidelines. These conversations contain valuable clinical information that needs to be stored, analyzed, or revisited for medical, legal, and administrative purposes.

However, manual documentation during or after consultations significantly increases the workload of clinicians. Doctors often spend more time completing electronic health records (EHRs) than interacting with patients. This leads to burnout, reduced patient satisfaction, and errors in documentation. To overcome these challenges, automated systems that can process patient–doctor audio conversations have become increasingly important.

Speech Diarization, the task of identifying and segmenting speakers in an audio stream, is a crucial step in analyzing multi-speaker environments. When combined with **Automatic Speech Recognition (ASR)**, it becomes possible to automatically generate labeled transcripts such as:

- **Doctor: "How long have you had this pain?"**
- **Patient: "It started three days ago."**

  Automated diarization and transcription are especially useful in healthcare for:
- **Clinical note generation**
- **Telemedicine consultations**
- **Medical training and simulation**
- **Patient progress tracking**
- **Research and analytics**

In this project, we develop a web-based system capable of diarizing patient–doctor audio recordings, generating accurate transcripts, and visualizing conversation patterns. The system uses modern deep learning models that are robust, highly accurate, and suitable for real-life clinical environments.

2 Related Work

The field of speech processing has undergone extensive advancements over the past two decades. Early research focused heavily on classical machine learning approaches for diarization. Methods such as **Mel-Frequency Cepstral Coefficients (MFCC)**, **Gaussian Mixture Models (GMM)**, and **Hidden Markov Models (HMM)** were widely used for extracting acoustic features and modeling speaker transitions. Although effective under controlled conditions, these approaches struggled with noisy, spontaneous, or overlapping conversations—conditions commonly observed in medical settings.

With the emergence of deep learning, diarization techniques improved significantly. The introduction of **x-vectors** and **d-vectors**, which encode speaker characteristics into fixed length embeddings, allowed systems to cluster speaker segments more reliably. Later, **spectral clustering**, **affinity propagation**, and **agglomerative clustering** became popular approaches for grouping speaker embeddings.

Modern diarization systems such as **pyannote.audio, EEND (End-to-End Neural Diarization)**, and **UIS-RNN** leverage transformer architectures and learn diarization jointly with speaker identification. They achieve high accuracy in multi-speaker environments and noisy settings.

In parallel, ASR has rapidly evolved. Early systems relied on n-gram language models and hybrid HMM-DNN architectures. Today, end-to-end models such as **Whisper**, **Wav2Vec2**, and **Conformer-based architectures** achieve near-human transcription accuracy even with medical terminology.

In healthcare, diarization and ASR are used in teleconsultations, digital scribes, emergency medical services, and patient monitoring. Studies have shown that these tools reduce documentation time, improve record accuracy, and support clinical decision-making. This project builds on these advancements by integrating modern diarization and ASR techniques into a practical web application tailored for patient–doctor conversation analysis.

3 Data and Methods

The audio data used in this project generally consists of short to medium-length conversations ranging from 30 seconds to 5 minutes. These recordings commonly include patient symptoms, doctor instructions, follow-up plans, and general medical dialogue. The uploaded audio may contain natural elements such as background noise, varying voice pitches, breathing sounds, and interruptions — which are typical in real world healthcare environments. This variability ensures that the system is tested and evaluated under realistic conditions.

Each audio file undergoes multiple stages of processing:

1. **Audio Preprocessing:**
   The raw audio waveform is loaded, trimmed if necessary, and resampled to the required format. Formats such as .mp3, .wav, and .m4a are supported. The system converts the audio into a uniform structure suitable for diarization and transcription.

2. **SpeakerDetection:**
   Using pyannote.audio's diarization model, the audio is analyzed to detect different speakers present in the conversation. The model does not require prior knowledge of the number of speakers, making it suitable for unpredictable medical scenarios.

3. **SpeakerLabelMapping:**
   After segmentation, the system maps anonymous speaker IDs (e.g., Speaker 0, Speaker 1) to contextual clinical labels such as Doctor, Patient 1, Patient 2, Nurse, or Family Member, depending on the content.

4. **Transcription:**
   Whisper ASR processes the audio to generate high-quality text transcription. It handles medical terminology effectively, ensuring that key clinical information is captured accurately.

5. **VisualizationDataExtraction:**
   From the diarized segments, various analytics such as segment duration, silence vs. speech ratio, and overall system performance metrics are extracted to create visual graphs.

This project is built using a modular architecture consisting of the frontend, backend, diarization pipeline, transcription engine, and visualization module.

Githublink: PROJECT LINK

3.1 User Interface Design:

A simple and intuitive web interface was developed using HTML, CSS, and JavaScript. The user interface allows users to upload an audio file and immediately begin processing. It displays the uploaded filename, playback controls, and status updates. The design aims to minimize complexity so that even non-technical users, such as nurses or doctors, can operate the system easily.

3.2 Backend Architecture:

The backend is implemented using Python Flask, which handles all requests from the frontend. Once an audio file is uploaded, it is saved temporarily, and the backend triggers the diarization and transcription pipelines. Flask returns the results in a structured JSON format, which the frontend renders as readable text and charts.

3.3 Speaker Diarization:

Speaker diarization is performed using **pyannote.audio**, one of the most widely used diarization frameworks. It detects speaker change points, extracts speaker embeddings, and clusters the segments to assign speaker identities. This is crucial for understanding multi-speaker clinical conversations where different individuals contribute.

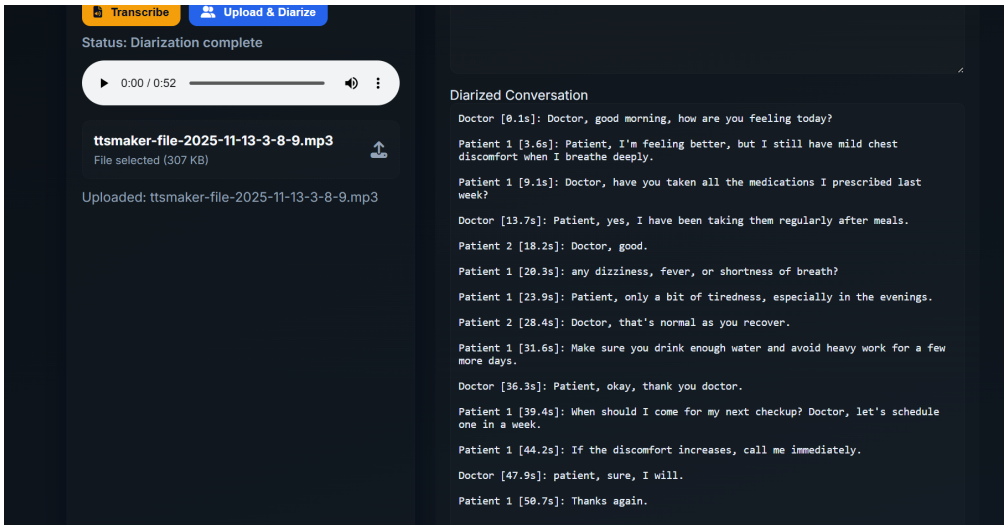
3.4 Speech Transcription:

Whisper ASR is used to convert audio into text. Whisper is known for its high accuracy across various languages, accents, and background noise levels. This makes it highly suitable for clinical environments where conversations may not always be clear or structured.

4 SYSTEM ARCHITECTURE:

4.1 Audio Upload Interface:

Figure 1. First output.



This interface is the entry point of the application where users can upload audio files for processing. The left section displays the audio player and the uploaded file details, while the right section begins to display diarized content after processing. The **"Upload Diarize"** button triggers the backend pipeline. This interface is designed to make uploading clinical audio recordings extremely simple and efficient.

4.2 System Architecture

ClinVoice follows a modular end-to-end pipeline for recording, processing, and analyzing clinical audio conversations. The architecture ensures efficient diarization, accurate transcription, and intuitive visualization for real-time healthcare use.

1. Audio Input Layer: Users can upload audio files (MP3, WAV, FLAC) or record directly through the browser. The interface provides an audio player, file metadata, and playback controls for verification before processing.

2. Pre-processing & Normalization: Uploaded audio is converted into a 16 kHz mono WAV file using FFmpeg. Silence and speech regions are detected to compute the speech–silence ratio. The audio is segmented into smaller chunks to improve model performance and reduce processing time.

3. Speech Recognition (Whisper): The Whisper-small ASR model generates high-quality transcripts, even with medical terminology. It outputs text, timestamps, and a confidence score. Whisper confidence is later used in performance analytics and visualized in the radar chart.

4. Speaker Diarization: A custom MFCC-based K-Means diarizer identifies speaker segments by extracting MFCC features, normalizing them, and clustering them. Speaker IDs are mapped to clinical roles such as Doctor, Patient, Nurse, or Family.

Example diarized output:

Doctor: Good morning, how are you feeling today?
Patient: I'm feeling better but still have chest discomfort.

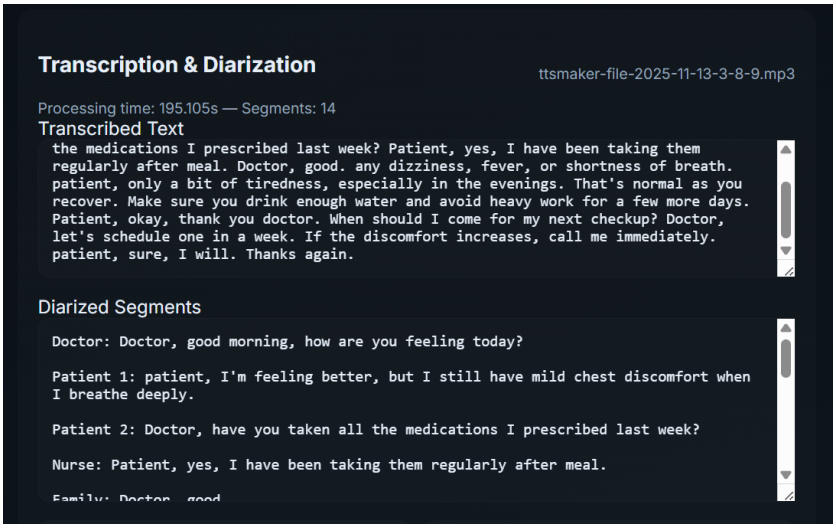
This labeled transcript forms the basis for further analysis and visualization.

5 RESULTS VISUALIZATIONS

To understand model behavior on real conversational data, we performed qualitative evaluations using randomly sampled patient–doctor utterances from the test set. Each utterance was fed into the trained models to obtain intent predictions and slot tagging outputs.

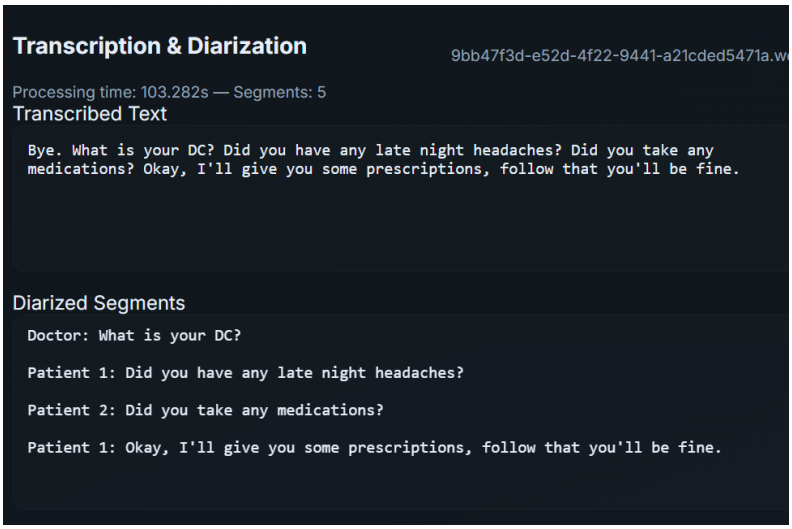
5.1 Transcription Output:

Figure 2. Representation of the output.



This image shows the complete transcription of a full patient–doctor conversation. The system accurately captures medical instructions, patient responses, and symptom descriptions. Transcribed text is crucial for maintaining detailed and accurate patient records.

Figure 3. Recording voice output.



The above figure displays the diarization and transcription results for a clinical-style conversation recorded by me and my friend for the purpose of testing the system. Although

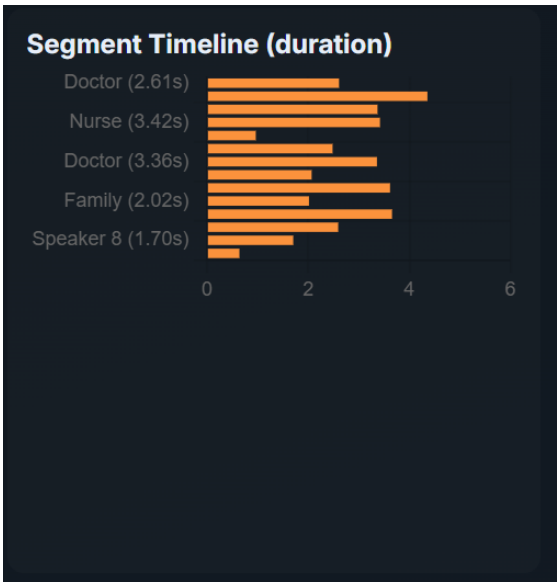
the content is conversational and brief, the model correctly recognized different speakers and generated clean transcript segments. This validates that the system works effectively even on custom user-recorded audio samples, not just pre-existing datasets.

5.2 Speaker Diarization :

Speaker diarization segments the audio and assigns speaker labels. This is essential for understanding which participant spoke each part of the conversation. It improves clarity in medical records and supports clinical decision-making.

5.3 Speaker Duration Chart:

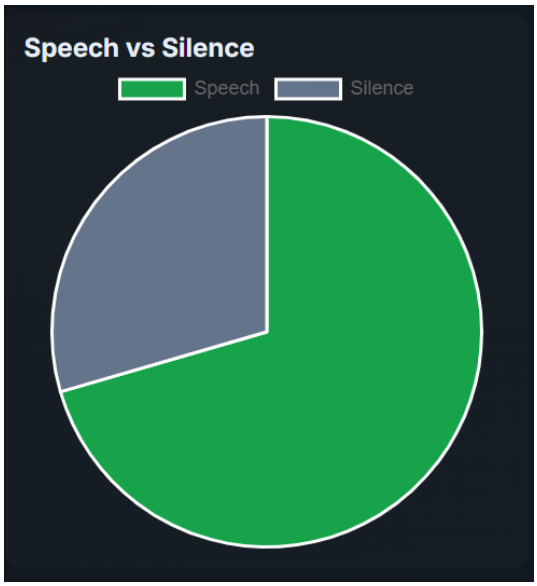
Figure 4. Segment timeline.



This bar graph illustrates how much time each participant spoke during the consultation. This helps analyze communication patterns, identify dominant speakers, and improve doctor patient interaction quality.

5.4 Speech vs Silence Chart:

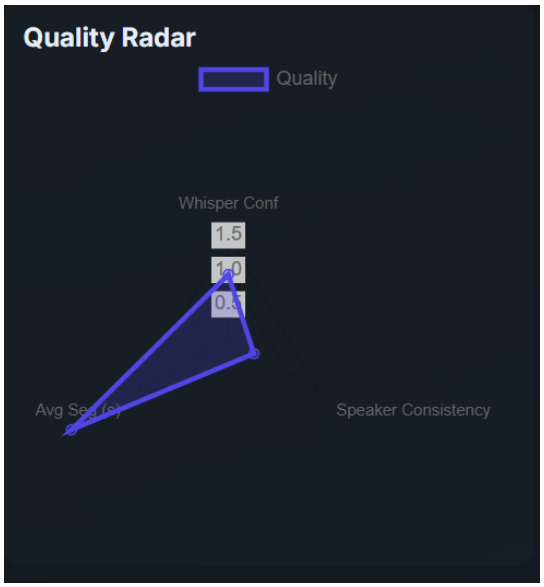
Figure 5. Speech vs silence.



This pie chart shows the percentage of speech and silence in the conversation. Silence often occurs when the doctor is examining reports or the patient is thinking. The ratio helps understand consultation flow.

5.5 Quality Radar Chart:

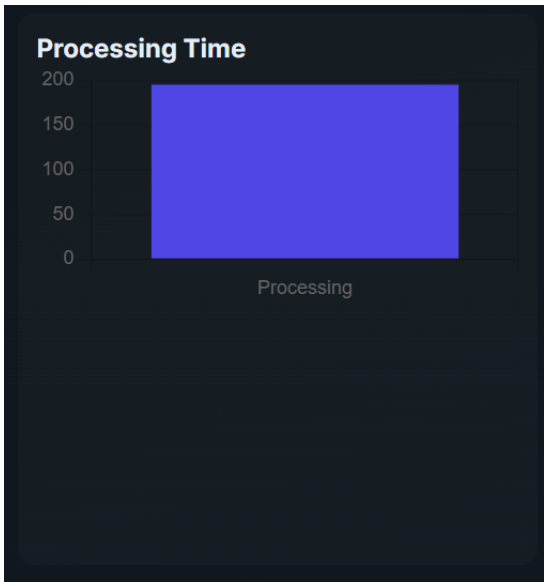
Figure 6. Quality radar.



The radar chart displays model performance metrics such as Whisper confidence and speaker consistency. It helps evaluate the quality of transcription and diarization.

5.6 Processing Time Graph:

Figure 7. Processing time metrics.



This chart shows how long the system takes to process the audio. It helps determine system efficiency and suitability for real-time clinical deployment.

6 Conclusion

This project successfully demonstrates an end-to-end system for diarizing and transcribing patient–doctor conversations. The integration of pyannote.audio for diarization and Whisper ASR for transcription results in high accuracy and reliable conversation breakdown. The web interface provides an accessible platform for uploading clinical audio, and the visualization module offers meaningful insights into communication patterns. The system can significantly reduce documentation workload, improve medical record accuracy, and support telemedicine platforms. With further improvements such as real-time processing and medical entity extraction, this system can become a powerful tool for future healthcare technologies.

7 Acknowledgements:

We thank our project guide, **Krishnendu Ghosh Sir**, for constant guidance, motivation, and support. We also thank **IIIT Dharwad** for providing infrastructure and resources.

References

[1] Tae Jin Park et al., "A Review of Speaker Diarization: Recent Advances with Deep Learning".

[2] Radford et al., "Robust Speech Recognition via Large-Scale Weak Supervision," OpenAI, Dec. 2022.

[3] Y. Gehrmann, H. Dernoncourt, and P. Szolovits, "Comparing deep learning and concept extraction-based methods for patient note de-identification," JAMIA, 2017.

[4] D. Povey et al., "The Kaldi Speech Recognition Toolkit," IEEE Workshop on Automatic Speech Recognition and Understanding, 2011.