Portable Transcriber with Advanced Speaker Role Identification

Tyler Rutley
Faculty Mentor: Kris Schindler
Department of Computer Science and Engineering
University at Buffalo
Buffalo, NY 142603
{tjrutley}@buffalo.edu

Abstract

This project demonstrates a real-time portable conversation-analysis system that incorporates streaming speech recognition, live speaker diarization, and continuous role inference using a locally hosted large language model. A Raspberry Pi equipped with a USB microphone and a character LED display streams audio to a PC server over a low-latency socket protocol. The server employs Speechmatics for word-level transcription and diarization, while an Ollama-hosted LLM asynchronously infers short semantic roles for each speaker as the conversation unfolds. Unlike prior LLM-based diarization methods, which operate offline and require full transcripts, this approach performs inference on finalized text segments, enabling responsive, on-device feedback in real time.
Early results show that combining traditional ASR diarization with LLM-driven semantic reasoning can enrich live transcripts with meaningful speaker-role annotations. The system highlights a middle ground between classical diarization pipelines and emerging LLM-based methods, offering a flexible foundation for future improvements.

1 Introduction

Understanding who is speaking and the role they play in a conversation is essential for interpreting interactions in domains such as customer service, healthcare, education, and assistive technologies. Traditional speaker diarization systems can identify "who spoke when," but often lack semantic understanding. Recent research shows that Large Language Models (LLMs) can infer speaker roles; however, these methods are typically used in an offline manner and require a full transcript. This limits their usefulness in real-time settings where immediate feedback is necessary.

This project addresses this gap by developing a portable, real-time conversation-analysis system that combines streaming speech recognition, live diarization, a USB microphone, and an LED display. The Raspberry Pi sends audio streams to a PC server over a low-latency socket. The server performs low-latency transcription and diarization, and in parallel updates each speaker's inferred role as the conversation unfolds. By integrating a traditional ASR pipeline with an LLM for semantic role identification, the system aims to provide improved conversational insight without the need for offline post

2 Related Works

Recent advances in speaker diarization and conversational analysis have increasingly incorporated LLMs to improve robustness and semantic understanding.

DiarizationLM by Wang et al. Wang et al. [2025] shows that LLMs can refine diarization outputs through post-processing, improving speaker-boundary accuracy once the full transcript is available. Likewise, Efstathiadis et al. Efstathiadis et al. [2024] propose a generalizable LLM-based correction framework that adjusts diarization labels using semantic reasoning. While both methods demonstrate strong improvements, they operate in an offline setting and require a full transcript before any inference can be made.

Commercial ASR systems such as Speechmatics Speechmatics [2024] provide high-quality streaming transcription with built-in diarization, enabling low-latency processing in practical deployments. Local LLM runtimes such as Ollama Ollama [2024] support on-device reasoning without relying on cloud APIs. Together, these tools form the foundation explored in this project.

3 Solution

The system is designed as a two-part architecture consisting of a portable Rapsberry Pi capture and display device, along with a PC-based processing server. Together, these two components enable real-time audio streaming, transcription, diarization, and LLM-based role inference.

3.1 Raspberry Pi Capture/Display Device

A Raspberry Pi Zero 2 W is equipped with a USB microphone and a 16x2 HD4478-compatible character LCD Hitachi [1998] serves as the front-end capture module. The device continuously records audio and transmits it to the server using a low-latency TCP socket. The pi itself, does not compute any of the speech processing. It simply captures the audio and streams it to the PC server. The LCD provides real-time display of what is said, which speaker ID, and the role the speaker plays in the conversation. The responsibilities of the Pi are summarized below:

- Capturing audio and forwarding it to the server
- Receiving processed metadata from the server
- Displaying real-time feedback on the character LCD

The LCD displays:

- The current transcription of the finalized text segment
- The speaker ID assigned by the diarization engine
- The role inferred via the LLM

This allows the Pi to work as a lightweight, portable visualization tool without the need for any local computation. Device configuration and remote access follow Raspberry Pi Foundation guidelines Raspberry Pi Foundation [2024].

3.2 PC Processing Server

The PC server performs all computationally intensive tasks. Incoming audio is processed by the Speechmatics streaming ASR engine Speechmatics [2024],which produces word-level transcriptions and speaker tags in real-time. Because Speechmatics provides both the partial and finalized transcription segments, the system can update the Pi display with minimal latency.

3.3 Role-Inference Prompting Strategy

The LLM is prompted using a fixed template designed to constrain the output to short, semantically meaningful role labels. The prompt instructs the model to infer a role for each speaker based solely on their own utterances and to return the result in a strict JSON structure. The exact prompt used is shown below:

```
You are analyzing a multi-speaker conversation.

Infer the likely role of EACH speaker that appears in the recent context.
```

```
Return no explanation and no text before or after the JSON structure.
If confidence is low, keep the role as "Unknown" (with high confidence).
If you can infer the name of the speaker, allow that to be the role.
Keep the role strictly 1-3 words.
Return ONLY valid JSON in this exact structure:

{
  "speakers": {
    "S1": [
      {"role": "Example Role 1", "confidence": 0.82}
    ],
    "S2": [
      {"role": "Example Role 2", "confidence": 0.77}
    ]
  }
}

Recent conversation context:
```

This prompt ensures that the model produces consistent, short role labels and adheres to a predictable output format, enabling reliable downstream parsing and real-time integration into the system.

3.4 Real-Time Data Flow

The system operates as a continuous real-time loop:

1. **Audio Capture:**
   The Raspberry Pi streams raw audio frames to the server over a persistent socket connection.
2. **Streaming ASR:**
   Speechmatics processes the incoming audio and emits both partial and finalized transcripts, each tagged with a speaker identifier.
3. **LLM Role Inference:**
   For every finalized segment, the server forwards the text and speaker ID to the LLM, which returns a short role label and confidence score.
4. **Metadata Packaging:**
   The server bundles the transcript segment, speaker ID, and inferred role into a compact message suitable for low-bandwidth transmission.
5. **LCD Display Update:**
   The Raspberry Pi receives the packaged message and updates the 16×2 character LCD to display:
   `[Speaker | Role]  segment_text`

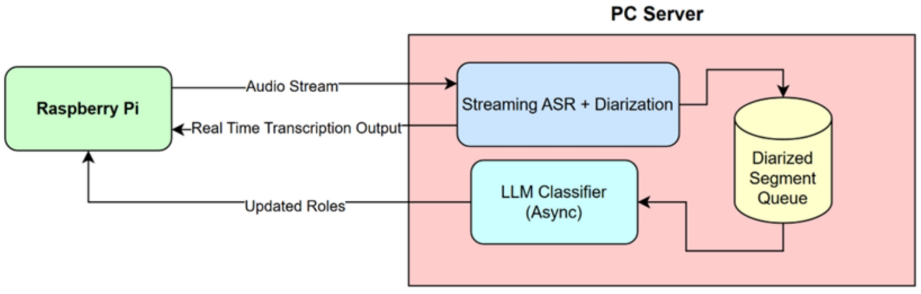

Figure 1: Overview of the real-time data flow between the Raspberry Pi and the server.

This pipeline enables low-latency, real-time conversational analysis on a portable device while offloading all computationally intensive processing to the server.

4 Demonstration Video

A demonstration video of the system working locally on the PC is available at the following link:

Real-Time System Demo (Video).

5 Limitations

Although the system successfully demonstrates real-time transcription and role inference on a portable platform, several limitations remain. First, the accuracy of speaker diarization can decrease in environments with significant background noise. Because the Raspberry Pi utilizes a lightweight USB microphone, noise can reduce the reliability of both the ASR output and the downstream role-inference performance.

Second, the role-classification process relies on short, finalized text segments. The lack of longer conversational memory restricts the stability of early role predictions. However, as the conversation unfolds, the model's confidence typically increases, producing more reliable results.

Third, the Raspberry Pi Zero 2 W introduces hardware constraints. Its limited processing power prevents any form of on-device audio processing, requiring all heavy computation to be offloaded to the server. This dependency shifts reliability to network latency, which cannot always be guaranteed. As a result, the system becomes susceptible to connection quality. The 16×2 LCD display also limits the amount of text that can be shown at once.

Finally, the system has not yet been evaluated across diverse accents, speaking styles, or multi-domain conversations. Variations in linguistic patterns may affect the accuracy and performance of transcription, diarization, and downstream role inference. Broader testing is needed to assess further generalization.

6 Future Work

Several methods can be explored to enhance the overall performance and autonomy of the system. First, incorporating personalized speaker profiles would allow the system to maintain persistent representations of recurring speakers. By storing voice embeddings or linguistic patterns, the system could stabilize identities across sessions and improve role consistency over longer conversations.

Second, adaptive on-device learning on the Raspberry Pi Zero 2 W could reduce reliance on the server for all processing. While the Pi cannot support the full ASR or LLM pipeline, lightweight models such as VAD, noise suppression, or incremental speaker-embedding refinement could improve audio quality and reduce the overall server load.

Third, handling overlapping speech remains a challenge in transcription systems. Integrating neural separation models or multi-channel diarization techniques could enable the system to identify and label overlapping utterances, improving both transcription quality and role inference.

Finally, spatial audio integration using multi-microphone arrays could allow the system to estimate speaker direction, thereby improving diarization accuracy. Additionally, directional visual feedback on the LCD could further assist users by indicating where each speaker's voice is originating.

7 Conclusion

This project demonstrates a practical and portable approach to real-time conversational analysis. It integrates streaming speech technologies with lightweight hardware and LLM-based semantic reasoning. By combining the Raspberry Pi with a server-side processing pipeline, the system delivers low-latency transcription, live speaker diarization, and continuous role inference without the need for offline post-processing. The use of the 16×2 LCD enables immediate, on-device visualization of

transcript segments, speaker identification, and inferred speaker roles, making the system accessible and responsive in real-world environments.

The results highlight the value of pairing traditional ASR technology with LLM-based semantic interpretation. While limitations remain, the system provides a strong foundation for future implementations. Overall, this work bridges the gap between offline LLM-based diarization research and real-time embedded applications.

References

G. Efstathiadis, V. Yadav, and A. Abbas. Llm-based speaker diarization correction: A generalizable approach. *arXiv preprint arXiv:2406.04927*, 2024. URL `https://arxiv.org/abs/2406.04927`.

Hitachi. *HD44780U: Dot Matrix Liquid Crystal Display Controller/Driver*, 1998. LCD controller datasheet. Commonly used in 16x2 character displays.

Ollama. *Ollama Documentation*, 2024. Retrieved from https://docs.ollama.com.

Raspberry Pi Foundation. *Remote Access Documentation*, 2024. Retrieved from https://www.raspberrypi.com/documentation/computers/remote-access.html.

Speechmatics. *Speechmatics Documentation*, 2024. Retrieved from https://docs.speechmatics.com/.

Q. Wang, Y. Huang, G. Zhao, E. Clark, W. Xia, and H. Liao. Diarizationlm: Speaker diarization post-processing with large language models. *arXiv preprint arXiv:2401.03506*, 2025. URL `https://arxiv.org/abs/2401.03506`.