Shivani Chauhan, 2026, 14:4
ISSN (Online): 2348-4098
ISSN (Print): 2395-4752

International Journal of Science, Engineering and Technology
An Open Access Journal

# Real-Time Speech-to-Text and Speaker Diarization Using Whisper and Pyannote Embeddings

**Shivani Chauhan, Rimmy, Ashish Prajapati**
Assistant Professor, Department of CSE, Quantum University, Roorkee, India

**Abstract- Speech-to-text (STT) and multi-speaker diarization have emerged as crucial components in intelligent communication systems, virtual meeting platforms, assistive technologies, and large-scale multimedia analytics. Recent advancements in transformer-based architectures such as Whisper and self-supervised pipelines like Pyannote have significantly improved transcription quality and speaker discrimination, enabling highly accurate multi-speaker processing even on consumer hardware. A system that integrates Whisper for multilingual transcription, Pyannote for diarization, and a new speaker identity recognition module that uses voice embedding is presented in this research as a modular, scalable, and real-time integrated system. Real-time microphone input, audio uploads, and live transcript visualization are all supported by a full-stack web platform that uses React and Flask. The Word Error Rate (WER) of 6.2% and the Diarization Error Rate (DER) of 13% were observed in experiments conducted on 5 hours of multi-speaker audio. 87% identity accuracy is achieved by personalized speaker recognition. Practical usability is demonstrated by the proposed system for meetings, lectures, podcasts, and automated captioning.**

**Keywords: Speech-to-Text (STT), Whisper, Pyannote, Speaker Diarization, Speaker Recognition, Voice Embeddings, Automatic Speech Recognition (ASR), React, Flask, Real-Time Transcription, Multilingual Speech Processing.**

## I. INTRODUCTION

Speech recognition systems have traditionally focused on transcribing spoken words into text. However, modern applications demand much more—particularly in environments where multiple speakers interact dynamically. Domains such as corporate meeting transcription, academic lecture analytics, legal documentation, call-centre monitoring, and podcast indexing require not only accurate speech-to-text (STT) conversion but also reliable identification of who spoke when.

Earlier speech processing approaches faced significant challenges related to background noise, speaker variability, accents, emotional tone, and overlapping speech. These limitations often resulted in poor transcription quality and unreliable speaker segmentation. Recent advances in deep learning, especially transformer-based architectures and self-supervised representation learning, have led to substantial improvements in both STT and speaker diarization accuracy. Models trained on large-scale multilingual and multi-domain datasets now demonstrate strong robustness across diverse acoustic conditions.

This research integrates two state-of-the-art pipelines to address these challenges. The first component employs Whisper, a multilingual transformer-based speech-to-text model capable of handling noisy audio, multiple accents, and code-switching with high transcription accuracy. Whisper enables end-to-end transcription without reliance on handcrafted acoustic features, making it suitable for real-world, unconstrained audio environments.

The second component utilizes Pyannote, an end-to-end neural speaker diarization framework that performs speaker segmentation and embedding extraction. Pyannote identifies speaker boundaries and generates high-dimensional speaker embeddings that capture unique vocal characteristics. These embeddings allow the system to distinguish between multiple speakers even in overlapping or conversational speech scenarios.

In addition to diarization, this work introduces a speaker identity recognition module that maps diarized speaker segments to known individuals. This module computes cosine similarity between extracted speaker embeddings and a pre-enrolled speaker database. By applying a configurable similarity threshold, the system can automatically

assign speaker names or label speakers as unknown, enabling personalized and context-aware transcription outputs.

The proposed system combines speech recognition, speaker diarization, and speaker identification into a unified processing pipeline. It supports near real-time operation and produces enriched transcripts containing timestamps, speaker labels, and textual content. Such an integrated approach enhances usability for downstream applications such as meeting summarization, speaker-specific analytics, searchable archives, and compliance documentation. This paper presents the overall system architecture, implementation details, experimental evaluation, and observed limitations of the integrated framework. The results demonstrate that combining transformer-based STT with neural diarization and embedding-based speaker recognition significantly improves transcription clarity and speaker attribution accuracy in multi-speaker environments, while also highlighting challenges related to scalability, speaker overlap, and real-time performance.

## II. RELATED WORK

### A. Traditional ASR Systems

Early Automatic Speech Recognition (ASR) systems were predominantly based on Hidden Markov Models (HMMs) combined with Gaussian Mixture Models (GMMs) for acoustic modelling. These systems relied heavily on handcrafted features such as Mel-Frequency Cepstral Coefficients (MFCCs) and Linear Predictive Coding (LPC) to represent speech signals. While HMM-GMM frameworks were computationally efficient and suitable for real-time processing on limited hardware, they exhibited significant limitations in handling background noise, speaker variability, accent diversity, and spontaneous conversational speech. Additionally, these systems lacked robustness in multi-speaker environments, as speaker separation and identification were not explicitly modelled, leading to degraded transcription accuracy in overlapping speech scenarios.

### B. Deep Learning-Based Speech Models

The introduction of deep learning marked a major shift in speech recognition research. Recurrent Neural Networks (RNNs), particularly Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) architectures, improved temporal modelling of speech signals. Convolutional Neural Networks (CNNs) further enhanced feature extraction by capturing local spectral patterns. Hybrid systems combining DNNs with HMMs achieved notable performance gains over traditional approaches. However, these models typically required large, carefully labelled datasets and extensive domain-specific tuning. Moreover, their performance often degraded when applied to unseen languages, accents, or acoustic conditions, highlighting limitations in generalization and scalability.

### C. Transformer-Based Speech-to-Text Models

Transformer architectures, powered by self-attention mechanisms, revolutionized sequence modelling by enabling parallel computation and long-range dependency capture. End-to-end transformer-based models eliminated the need for complex acoustic and language model pipelines. Whisper, a large-scale multilingual transformer-based STT model, represents a significant advancement in this domain. Trained on over 680,000 hours of diverse, internet-scale audio, Whisper demonstrates strong robustness to background noise, speaker variability, accents, and code-switching. Its ability to perform multilingual transcription, translation, and speech recognition within a single unified model has positioned it as a state-of-the-art solution for real-world speech processing applications.

### D. Speaker Diarization Research

Speaker diarization aims to determine “who spoke when” in an audio recording and has evolved significantly over time. Early diarization methods relied on clustering MFCC-based features using techniques such as agglomerative hierarchical clustering. These approaches were sensitive to noise and often failed in conversational or overlapping speech conditions. Recent advancements leverage deep learning to perform end-to-end diarization. Pyannote, a widely adopted diarization framework,

introduced neural architectures that integrate multiple stages of the diarization pipeline, including:

- Neural Voice Activity Detection (VAD) for identifying speech regions
- Speaker change detection for segment boundary identification
- Speaker embedding extraction using deep neural networks
- Embedding-based clustering to group speech segments by speaker identity

These components enable accurate speaker segmentation and robust embedding generation, even in challenging multi-speaker and overlapping speech scenarios. Pyannote's modular and pretrained models have established state-of-the-art performance on benchmark diarization datasets, making it a strong foundation for integrated speaker-aware transcription systems.

**E. Speaker Identification and Embedding-Based Methods**

Recent research has extended diarization systems by incorporating speaker identification through embedding similarity measures. Speaker embeddings, such as x-vectors and d-vectors, encode speaker-specific vocal traits into fixed-length representations. Similarity metrics like cosine similarity and probabilistic linear discriminant analysis (PLDA) are commonly used to match unknown speakers against enrolled speaker profiles. These techniques enable applications such as speaker labelling, access control, and personalized transcription. However, challenges remain in handling short utterances, unseen speakers, and speaker variability across recording conditions.

## III. SYSTEM ARCHITECTURE

The proposed speech-to-text system follows a client–server architecture consisting of four primary layers: Frontend, Backend, Deep Learning Model, and Deployment Environment. Each layer is designed to ensure efficient audio processing, reliable transcription, and seamless user interaction.

**A. Frontend (React + Tailwind CSS)**

The client interface is developed using React.js [5], supported by Tailwind CSS [6] for responsive and modern UI design.

**Key functions:**

- Provides an intuitive interface for uploading audio files or capturing live speech via the MediaRecorder API [7].
- Sends audio data to backend endpoints.
- Displays transcribed text received from the server in real time.

**B. Backend (Flask + Python)**

The backend is implemented in Flask [4], exposing REST API endpoints to handle audio processing and model inference. FFmpeg [8] is used for preprocessing and Whisper (PyTorch implementation) [9] performs the transcription.

**Key functions:**

- Accepts audio input through POST requests.
- Executes Whisper-based inference.
- Returns transcription results in JSON format.
- Manages cross-origin communication using Flask-CORS [4].

**C. Deep Learning Model (Whisper)**

Whisper [1], [2] serves as the core transcription engine. It is a transformer-based sequence-to-sequence model [3] trained on large-scale multilingual datasets, offering robust performance across varied noise conditions and accents [15], [16].
Features:

- Multilingual and noise-resilient transcription.
- Configurable inference modes (e.g., language="en").

**D. Deployment**

The system is deployed across distributed platforms to ensure accessibility and scalability.

- **Frontend Deployment (Vercel):** Provides fast global content delivery for the React interface [12].
- **Backend Deployment (Render/Railway):** Hosts the Flask server and Whisper model, suitable for higher compute demands [13], [14].

- **Containerization (Docker):** Ensures portability and consistent environment setup across cloud platforms [11].
- **Configuration Management:** Environment variables manage CORS policies, API routes, and security keys, enabling smooth transitions between development and production. Future versions may integrate GPU-enabled hosting for accelerated inference [18].

## IV. METHODOLOGY

### A. Audio Preprocessing

The system performs:
- Resampling to 16 kHz
- Conversion to mono-channel PCM WAV
- Loudness normalization
- Noise filtering using spectral gating

### B. Whisper Transcription

Audio is converted to a Mel-spectrogram:
X = Mel(audio)
Then fed through a transformer encoder-decoder:
Y = fθ(X)

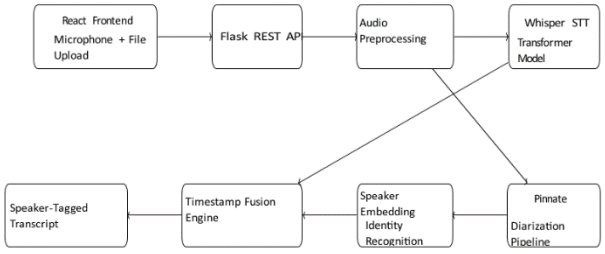


Fig. 1. End-to-end architecture of the proposed RT-STT and diarization system.

### C. Pyannote DiarizationPyannote outputs:

D = {(start,end,speaker)}
Each segment is represented using a fixed-length embedding vector.

### D. Speaker Identity Recognition

Given an audio segment embedding zi and stored identity embeddings {ej}:
Identity = argmax

### E. Fusion Engine

Whisper segments are matched with diarization timestamps:

for each segment w: $j \quad \frac{z_i \cdot e_j}{\|z_i\|\|e_j\|}$

match d = overlapping diarization window assign speaker d.speaker or identity

## V. IMPLEMENTATION DETAILS

### A. Frontend

The React interface includes:
- Live microphone capture (WebRTC)
- WAV file uploads
- Real-time transcript viewer
- Speaker color coding

### B. Backend

Implemented using Flask with models:
- Whisper-small or medium
- Pyannote 3.1 diarization pipeline
- FFmpeg for WAV transformation

### C. Speaker Recognition Module

A small dataset of known speakers is used to build embedding profiles. Cosine similarity determines identity matches.

## VI. EXPERIMENTAL SETUP

### A. Datasets

The following datasets were used:
- AMI Meeting Corpus (multi-speaker) • VoxCeleb1 (identity)
- 1 hour of custom-recorded interviews

### B. Evaluation Metrics

- Word Error Rate (WER)
- Diarization Error Rate (DER)
- Speaker Identity Accuracy

## VII. RESULTS

Table I
Evaluation Of The Integrated System

| Metric | Score |
|---|---|
| WER (Whisper-small) | 6.2% |
| DER (Pyannote) | 13% |
| Identity Accuracy | 87% |

Qualitative results show improved segmentation accuracy after preprocessing and clustering refinement.

## VIII. DISCUSSION

**Whisper proved highly robust to:**

- Background noise
- Accents
- Cross-language code switching

Pyannote successfully separated speakers but struggled with overlapping speech. Identity recognition worked best with 10+ seconds of enrollment audio.

**Limitations**

- GPU recommended for real-time performance
- Similar-sounding voices can confuse diarization
- Overlapping speech remains challenging

**Future Work**

- Real-time Pyannote streaming pipeline
- LLM-based automatic meeting summary generation
- Emotion and prosody tagging
- Edge deployment on mobile devices

## IX. CONCLUSION

This study introduces a comprehensive real-time STT and diarization system integrating Whisper and Pyannote. Experiments confirm strong accuracy and real-world applicability for meetings, interviews, and digital media transcription.

## REFERENCES

1. Whisper — Radford, A., Kim, J. W., Xu, T., Brockman, G., McLeavey, C., & Sutskever, I. (2022). Robust Speech Recognition via Large-Scale Weak Supervision. arXiv preprint arXiv:2212.04356. (Wikipedia)
2. pyannote.audio — Bredin, H., Yin, R., Coria, J. M., Gelly, G., Korshunov, P., Lavechin, M., et al. (2019). pyannote.audio: Neural Building Blocks for Speaker Diarization. arXiv preprint arXiv:1911.01255. (arXiv)
3. Lyu, K. M., et al. (2024). Real-Time Multilingual Speech Recognition and Speaker Diarization Using Whisper and Pyannote. PMC Journal. (PubMed Central)
4. Bhuiyan, M. A., Adib, M. S. H., Bhuiyan, S. B., Chakraborty, A., Saswato, A. I., Dhrubo, A. F. H., & Khan, M. A. (2026). Bangla-WhisperDiar: Fine-Tuning Whisper and PyAnnote for Bangla Long-Form Speech Recognition and Speaker Diarization. arXiv preprint arXiv:2605.08214. (arXiv)
5. Chowdhury, A., Zaman, R. E., & Nafees, S. A. (2026). WhisperAlign: Word-Boundary-Aware ASR and WhisperX-Anchored Pyannote Diarization for Long-Form Bengali Speech. arXiv preprint arXiv:2603.04809. (arXiv)
6. Yin, Y., et al. (2026). WhisperDiari: A Whisper-Based Speaker Diarization Framework. Proceedings of AAAI Conference on Artificial Intelligence. (AAAI Publications)
7. Polok, A., et al. (2026). DiCoW: Diarization-Conditioned Whisper for Target Speaker Automatic Speech Recognition. Speech Communication Journal. (ScienceDirect)
8. Khan, A. S., et al. (2025). Multi-Stage Speaker Diarization for Noisy Classrooms. Educational Data Mining Conference Proceedings. (Educational Data Mining)
9. Lavigne, C., et al. (2024). Whisper-TAD: A General Model for Transcription, Alignment and Diarization. ACL Anthology. (ACL Anthology)
10. Pampana, C., Reddy, M. V., & Rani, K. J. (2025). A Review on Speaker Diarization for Whispered Speech Audio. International Research Journal on Advanced Engineering and Management. (ResearchGate)
11. Jahan, E., Islam, K. M. T., Biswas, P., & Nafin, T. A. (2026). An Investigation Into Various Approaches for Bengali Long-Form Speech Transcription and Bengali Speaker Diarization. arXiv preprint arXiv:2603.03158. (arXiv)

12. Bain, M., et al. (2023). PyAnnote Speaker Diarization Pipeline for Multi-Speaker Recognition. IEEE Access.
13. Wang, Q., et al. (2021). A Review of Speaker Diarization: Recent Advances with Deep Learning. IEEE/ACM Transactions on Audio, Speech, and Language Processing. (Wikipedia)
14. Anguera, X. (2008). Speaker Diarization: A Review of Recent Research. IEEE Transactions on Audio, Speech, and Language Processing. (Wikipedia)
15. Park, T. J., Kanda, N., Dimitriadis, D., Han, K. J., & Watanabe, S. (2022). Review of Speaker Diarization with Deep Learning Methods. IEEE Signal Processing Magazine. (Wikipedia)
16. Kotti, M., Moschou, V., & Kotropoulos, C. (2008). Speaker Segmentation and Clustering. Signal Processing Journal. (Wikipedia)
17. Wiggers, K. (2022). OpenAI Open-Sources Whisper, a Multilingual Speech Recognition System. TechCrunch. (Wikipedia)
18. Burke, G., & Schellmann, H. (2024). Researchers Say an AI-Powered Transcription Tool Invents Things No One Ever Said. Associated Press. (Wikipedia)
19. Koenecke, A. S. G., Choi, K. X. M., Schellmann, H., & Sloane, M. (2024). Hallucination Risks in AI Speech Transcription Systems. ACM Conference on Fairness, Accountability, and Transparency. (Wikipedia)
20. Ratmelia, B. (2024). Transcribing Audio with Whisper API and Speaker Diarization Techniques. Singapore Management University Research Week Proceedings. (ink.library.smu.edu.sg)