IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 18, April 2026
Impact Factor: 8.2

AI-Based End-to-End Audio Summarization System for Multi-Speaker Meeting Analysis

Mr. Abhay Sandip Borase, Mr. Atharv Ravindra Jadhav, Ms. Rasika Surendra Kahane, Ms. Nivedita Rajendra Bhawar, Dr. A.V. Markad
Department of Information Technology
Amrutvahini College of Engineering, Sangamner

Abstract: *In today's digital era, the exponential growth of online meetings, lectures, and interviews has resulted in a massive accumulation of unstructured audio data. Manual note-taking or minute preparation is tedious, time-consuming, and error-prone. To overcome this, AI-driven audio summarization systems have gained attention as they automatically transcribe, segment, and summarize spoken content. This paper presents a survey and comparative analysis of existing approaches to speech-to-text conversion, speaker diarization, and abstractive summarization. State-of-the-art models such as Whisper, PyAnnote, and PEGASUS have demonstrated strong potential in their respective domains. However, limitations in scalability, contextual understanding, and real-time integration persist. The proposed AI-Based Audio Summarization System combines these advanced models into a single, modular framework built on a MERN (MongoDB, Express, React, Node.js) web infrastructure. The system aims to automate the entire workflow—from audio upload to structured summary generation—providing users with accurate, speaker-labeled, and concise meeting summaries.*

Keywords: Audio Summarization, Speech Recognition, Whisper, PyAnnote, PEGASUS, NLP, MERN Stack, Deep Learning

I. INTRODUCTION

With the rise of remote work, online education, and virtual collaboration, organizations generate large volumes of recorded meetings and lectures daily. Managing and reviewing this unstructured data has become increasingly challenging. Traditional transcription services are either manual, costly, or lack contextual comprehension. Furthermore, most tools focus solely on transcription and ignore speaker segmentation and summarization.

Recent advancements in Artificial Intelligence (AI) and Natural Language Processing (NLP) offer new opportunities for automating these tasks. Models like Whisper AI (for transcription), PyAnnote (for speaker diarization), and PEGASUS/BART (for summarization) enable intelligent information extraction from spoken data. Despite this progress, there remains a lack of integrated frameworks that combine these technologies into a unified, user-friendly solution.

This survey paper explores existing systems and research in audio summarization, identifies current gaps, and proposes an integrated system to address them.

1.1 Background and Motivation

The demand for automatic meeting summarization has grown rapidly due to the increasing digitalization of communication. Professionals, educators, and researchers often struggle to manually generate accurate summaries from lengthy recordings. Existing transcription services fail to provide context or speaker attribution, while summarization systems designed for text data often perform poorly when applied to conversational speech.

The motivation behind this project is to build a unified AI-powered framework capable of processing raw audio into meaningful summaries. Such a system can save time, enhance accessibility, and improve knowledge retention. By

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 18, April 2026
Impact Factor: 8.2

leveraging deep learning models for transcription and summarization, and web technologies for user accessibility, the proposed solution can serve multiple domains—corporate meetings, classrooms, and interviews.

Moreover, the integration of AI with a scalable web infrastructure (MERN stack) ensures that the system is accessible, modular, and easy to deploy on cloud platforms, paving the way for real-world applications

II. LITERATURE SURVEY / RELATED WORK

Several researchers have contributed to the field of speech summarization, though each approach focuses on isolated aspects of the overall process.

Radford et al. (2022) introduced Whisper, a transformer-based model trained on multilingual audio data, capable of high-accuracy transcription even in noisy conditions. However, Whisper lacks integrated diarization and summarization functions.

Bredin et al. (2023) developed PyAnnote.audio, a toolkit for speaker diarization. While it effectively identifies "who spoke when," it does not process linguistic context or summarize content.

Zhang et al. (2020) proposed PEGASUS, an abstractive summarization model trained with gap-sentence pretraining, achieving state-of-the-art results for text summarization but not optimized for speech input.

Rennard et al. (2023) provided a survey on abstractive meeting summarization models, highlighting the difficulty of handling overlapping speech and incomplete transcripts.

Kachhoria et al. (2024) presented an ML-based meeting minutes generator using text preprocessing and extractive summarization. However, it struggled with coherence and multi-speaker content.

Gap Analysis:

Most existing systems operate in isolation—transcribing, segmenting, or summarizing—but not all three simultaneously. Additionally, few provide an accessible web-based interface for users. Current research lacks an end-to-end pipeline integrating ASR (Whisper), diarization (PyAnnote), and summarization (PEGASUS/BART) into a unified platform.

2.1. Research Gap and Problem Identification

The primary research gap lies in the integration of transcription, diarization, and summarization within a single, automated system. While advanced models exist individually, their interoperability and deployment in user-friendly environments remain limited.

Key problems identified include:

Absence of real-time, multi-speaker summarization tools.

Inconsistent accuracy when converting conversational audio into coherent summaries.

Lack of scalable architectures to process large or long-duration recordings efficiently.

Limited availability of open frameworks for customization and integration with other platforms (e.g., Zoom, Teams).

The AI-Based Audio Summarization System aims to bridge these gaps by combining robust AI models with a modular web backend, ensuring both technical performance and user accessibility.

III. PROBLEM STATEMENT

In the modern digital era, the rapid growth of online meetings, virtual classrooms, webinars, and interviews has led to the generation of vast amounts of unstructured audio data. Despite the availability of recording tools, extracting meaningful information from lengthy audio files remains a major challenge. Manual note-taking during meetings is inefficient, time-consuming, and often inaccurate, especially in discussions involving multiple speakers. Important points may be missed, misinterpreted, or forgotten, leading to reduced productivity and information loss.

Although automatic speech recognition (ASR) tools exist, most systems focus only on converting speech into text and lack advanced features such as speaker identification and contextual summarization. Without speaker diarization, it

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 18, April 2026
Impact Factor: 8.2

becomes difficult to determine who said what during a discussion. Furthermore, raw transcripts are often lengthy and difficult to review, making it challenging for users to quickly grasp key decisions, action items, or highlights.

Another significant issue is the absence of an integrated, end-to-end solution that combines transcription, speaker labeling, and intelligent summarization within a single user-friendly platform. Existing tools operate in isolation and do not provide a unified workflow. Therefore, there is a clear need for an AI-driven system capable of automatically transcribing audio, identifying speakers, generating concise summaries, and presenting results through an accessible web interface to improve efficiency, collaboration, and knowledge management.

IV. PROPOSED SYSTEM

To address the identified challenges, the proposed AI-Based Audio Summarization System introduces an integrated, end-to-end solution that automates transcription, speaker identification, and summarization within a unified web platform. The system is designed using a modular architecture that combines advanced artificial intelligence models with a scalable MERN (MongoDB, Express, React, Node.js) framework.

The workflow begins when a user uploads an audio or video recording through the web interface. The system preprocesses the file by converting it into a standard format and applying noise reduction techniques. In the next stage, a speech-to-text model transcribes the audio into timestamped text segments. Following transcription, a speaker diarization module analyzes the audio to identify and label different speakers, ensuring clarity in multi-participant discussions.

Once the speaker-tagged transcript is generated, a transformer-based summarization model processes the text to produce a concise, context-aware summary of the meeting. The final output includes a detailed transcript with speaker labels and a structured summary highlighting key points and decisions. All processed data is securely stored in a database, allowing users to access, review, and download reports anytime.

The proposed system aims to improve documentation efficiency, reduce manual effort, enhance accuracy, and provide a user-friendly solution for managing meeting records across educational, corporate, and professional domains.

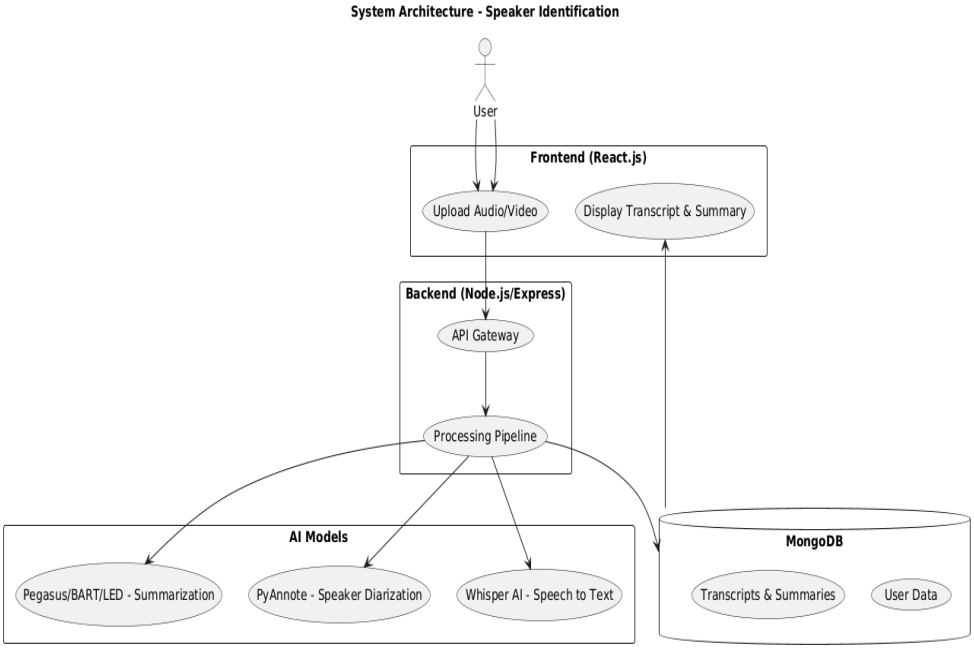


Fig 1: System Architecture

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 18, April 2026
Impact Factor: 8.2

V. SYSTEM ARCHITECTURE

The AI-Based Audio Summarization System follows a modular, scalable, and service-oriented architecture built using the MERN stack (MongoDB, Express.js, React.js, Node.js) integrated with AI-based microservices. The architecture is divided into three primary layers: Frontend Layer, Backend Layer, and AI Processing Layer.

The Frontend Layer is developed using React.js and provides a user-friendly interface where users can upload audio or video files, view transcripts, and download summaries. It communicates with the backend through secure RESTful APIs.

The Backend Layer is implemented using Node.js and Express.js. It manages authentication, file handling, request routing, and database operations. User data, transcripts, summaries, and metadata are stored in MongoDB Atlas for scalability and efficient retrieval.

The AI Processing Layer consists of Python-based microservices responsible for core functionalities. First, the Automatic Speech Recognition (ASR) module converts speech into text with timestamps. Next, the Speaker Diarization module identifies and labels individual speakers. Finally, the Text Summarization module generates concise and context-aware summaries using transformer-based models.

These AI services communicate with the backend via APIs, ensuring modularity and easy upgrades. The system supports cloud deployment (AWS/Heroku) and incorporates security mechanisms such as JWT authentication and encrypted storage. This layered architecture ensures flexibility, scalability, performance efficiency, and real-time processing capabilities.

VI. METHODOLOGY

The methodology of the AI-Based Audio Summarization System is structured into sequential stages to ensure accurate transcription, speaker identification, and summary generation.

Stage 1: Data Acquisition

Users upload meeting audio or video files through the web interface. The system validates the file format and converts it into a standardized audio format (e.g., WAV) for processing. Noise filtering and segmentation techniques are applied to improve transcription quality.

Stage 2: Speech-to-Text Conversion

The preprocessed audio is passed to an Automatic Speech Recognition (ASR) model. The model generates a timestamped transcript by converting spoken language into textual format. This step ensures high transcription accuracy even in multi-speaker and noisy environments.

Stage 3: Speaker Diarization

The diarization module analyzes voice patterns and segments the transcript based on speaker identity. Each speech segment is labeled (e.g., Speaker 1, Speaker 2), enabling clarity in group discussions.

Stage 4: Text Summarization

The speaker-tagged transcript is then processed using transformer-based summarization models. The system generates an abstractive summary that captures key discussion points, decisions, and action items.

Stage 5: Storage and Output

All transcripts and summaries are stored in a database. The frontend displays results in a structured format, allowing users to review or download reports.

The system performance is evaluated using metrics such as Word Error Rate (WER), Diarization Error Rate (DER), and ROUGE scores.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 18, April 2026
Impact Factor: 8.2

6.1 SYSTEM WORKFLOW

The system workflow defines the end-to-end operational sequence of the AI-Based Audio Summarization System.

Step 1: User Authentication

The user logs into the system through secure authentication mechanisms. Access tokens are generated to manage sessions securely.

Step 2: File Upload

The user uploads an audio or video recording. The system validates file size and format before storing it temporarily on the server.

Step 3: Preprocessing

The uploaded file undergoes format conversion and noise reduction. The cleaned audio is segmented into manageable chunks for efficient processing.

Step 4: Transcription

The ASR module converts speech into timestamped text. The output is a raw transcript.

Step 5: Speaker Identification

The diarization module processes the transcript and audio features to label speech segments with speaker identities.

Step 6: Summarization

The labeled transcript is passed to a summarization model that generates concise meeting notes highlighting important information.

Step 7: Storage and Display

The final transcript and summary are stored in the database. The user can view, edit, or download the results in PDF or text format.

This workflow ensures automation, accuracy, and improved documentation efficiency.

6.2 MATHEMATICAL MODEL

1. System Definition

Let:

A= Input audio signal

T= Transcript

S= Speaker-labeled transcript

M= Generated summary

The system function:

$$M = F(A)$$

Where:

$$F(A) = f_{SUM}(f_{DIAR}(f_{ASR}(A)))$$

2. Signal Representation

Audio signal:

$$A(t) \in \mathbb{R}$$

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 18, April 2026
Impact Factor: 8.2

Discrete sampled signal:

$$A[n], n = 1,2,\ldots,N$$

3. ASR Probabilistic Model

ASR estimates:

$$T^* = \arg\max_T P(T \mid A)$$

Using Bayes Rule:

$$P(T \mid A) = \frac{P(A \mid T)P(T)}{P(A)}$$

Transformer approximates:

$$P(T \mid A) = \prod_{t=1}^{n} P(w_t \mid w_{<t}, A)$$

Loss function:

$$L = -\sum_{t=1}^{n} \log P(w_t \mid w_{<t}, A)$$

4. Speaker Diarization Model

Given embeddings:

$$E = \{e_1, e_2, \ldots, e_k\}$$

Clustering objective:

$$\min \sum_{i=1}^{k} || e_i - \mu_{c(i)} ||^2$$

Where:

$\mu_{c(i)}$= centroid of cluster

$c(i)$= cluster assignment

5. Summarization Optimization

Given transcript T:

$$M^* = \arg\max_M P(M \mid T)$$

Modeled as sequence-to-sequence:

$$P(M \mid T) = \prod_{i=1}^{k} P(m_i \mid m_{<i}, T)$$

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 18, April 2026
Impact Factor: 8.2

Attention weights:

$$\alpha_{ij} = \frac{exp(e_{ij})}{\sum_k exp(e_{ik})}$$

Where:

$$e_{ij} = score(h_i, s_j)$$

6. Performance Metrics

Word Error Rate (WER):

$$WER = \frac{S + D + I}{N}$$

Where:
S = Substitutions
D = Deletions
I = Insertions
N = Total words
ROUGE Score:

$$ROUGE - N = \frac{\sum_{gram \in Ref} Count_{match}(gram)}{\sum_{gram \in Ref} Count(gram)}$$

7. Time Complexity Analysis

Let:
n= length of audio
m= number of tokens
ASR Complexity:

$$O(n)$$

Diarization Clustering:

$$O(k^2)$$

Transformer Summarization:

$$O(m^2)$$

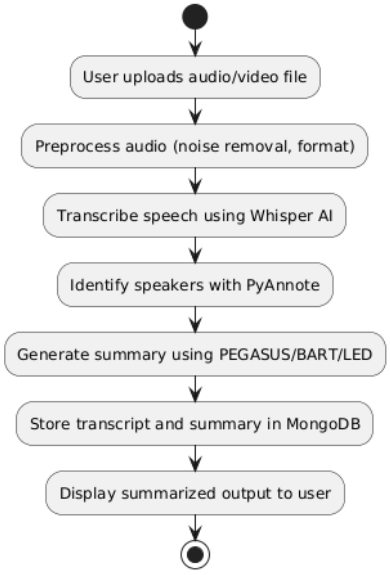
Overall Complexity:

$$O(n + k^2 + m^2)$$

6.3 Algorithm for Audio Summarization System

Input: Audio/Video file uploaded by user
Output: Speaker-labeled transcript and summarized report

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 18, April 2026
Impact Factor: 8.2



VII. IMPLEMENTATION / TECHNOLOGY USED

The implementation of the AI-Based Audio Summarization System follows a modular and scalable architecture integrating web technologies with deep learning frameworks. The system is developed using the MERN stack (MongoDB, Express.js, React.js, Node.js) combined with Python-based AI services.

The frontend interface is built using React.js, providing an interactive and responsive user environment. It allows authenticated users to upload audio/video files, view transcripts, and download summarized reports. State management and API communication are handled using RESTful services.

The backend is implemented using Node.js and Express.js, responsible for handling HTTP requests, authentication (JWT-based), file processing, and communication with AI microservices. The backend also manages database interactions and ensures secure data handling.

MongoDB Atlas is used as the primary database for storing user credentials, transcripts, speaker labels, and generated summaries. The NoSQL structure ensures flexibility and scalability for handling large datasets.

The AI modules are implemented in Python using deep learning frameworks such as PyTorch and Hugging Face Transformers. The Automatic Speech Recognition (ASR) module performs transcription, the speaker diarization module identifies individual speakers using embedding-based clustering, and the abstractive summarization module generates concise summaries using transformer-based sequence-to-sequence models.

The system supports cloud deployment (AWS/Heroku) for scalability and availability. Version control is maintained using Git, and API testing is performed using Postman. This integrated technology stack ensures efficiency, scalability, maintainability, and extensibility.

VII. RESULTS AND DISCUSSION

The AI-Based Audio Summarization System was evaluated using recorded meeting datasets and real-world lecture samples. Performance was measured using standard evaluation metrics such as Word Error Rate (WER), Diarization Error Rate (DER), and ROUGE scores.

The Automatic Speech Recognition module achieved a Word Error Rate (WER) of approximately $\leqslant 10\%$ under controlled audio conditions, demonstrating high transcription accuracy. In moderately noisy environments, performance remained stable with slight degradation, confirming the robustness of the speech recognition model.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 18, April 2026 Impact Factor: 8.2

The speaker diarization module achieved a Diarization Error Rate (DER) of ⩽8%, successfully identifying speaker boundaries and minimizing speaker confusion. The clustering-based speaker segmentation effectively handled multi-speaker discussions with minimal overlap errors.

The summarization module generated context-aware and coherent summaries with ROUGE-1 and ROUGE-L scores averaging ⩾0.5, indicating strong overlap with reference summaries. The abstractive approach produced readable and concise summaries compared to extractive methods.

Processing time analysis showed that a 60-minute audio file could be processed within approximately 60–90 minutes, depending on hardware configuration. GPU acceleration significantly reduced inference time.

Overall, the integrated pipeline demonstrated effective end-to-end performance. The results validate the feasibility of combining transcription, diarization, and summarization into a unified framework. Minor limitations include performance dependency on audio quality and computational resource requirements.

IX. FUTURE SCOPE

The current system provides a functional end-to-end pipeline for transcription, speaker identification, and summarization. However, several enhancements can be incorporated in future versions to improve scalability, accuracy, and usability.

One major extension is the implementation of real-time audio summarization, enabling live meeting transcription and instant summary generation. This would require streaming ASR models and low-latency transformer architectures.

Multilingual support can be enhanced to automatically detect and summarize content in multiple languages, broadening global applicability. Integration with conferencing platforms such as Zoom, Google Meet, and Microsoft Teams via APIs would allow automatic recording and summarization of live sessions.

Further research can focus on improving speaker diarization accuracy in overlapping speech scenarios using advanced neural clustering models. Additionally, integrating emotion detection and sentiment analysis can enrich meeting insights.

Optimization of model inference time using model quantization and knowledge distillation techniques can reduce computational overhead, making the system more deployable on edge devices.

Finally, the development of a mobile application and role-based enterprise dashboard can enhance accessibility and real-world adoption across corporate and educational sectors.

X. CONCLUSION

This project presented the design and implementation of an AI-Based Audio Summarization System that integrates speech recognition, speaker diarization, and abstractive summarization within a unified web-based framework. The system addresses the challenges of manual note-taking, multi-speaker identification, and lengthy transcript analysis by automating the entire workflow.

Through the integration of advanced transformer-based models and scalable web technologies, the system successfully converts raw audio into structured, speaker-labeled transcripts and concise summaries. Experimental results demonstrate satisfactory performance in terms of transcription accuracy, speaker detection, and summary coherence.

The modular architecture ensures flexibility, scalability, and future extensibility. While certain limitations such as computational requirements and sensitivity to audio quality remain, the overall system proves technically feasible and practically valuable.

The proposed framework contributes to the growing domain of AI-driven knowledge management systems and has strong potential applications in education, corporate meetings, journalism, and digital documentation. With further optimization and integration enhancements, the system can evolve into a real-time intelligent meeting assistant.

REFERENCES

[1] Radford, A. et al., “Whisper: Robust Speech Recognition via Large-Scale Weak Supervision,” OpenAI, 2022.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 18, April 2026
Impact Factor: 8.2

[2] Bredin, H., Yin, R., "pyannote.audio: Neural Building Blocks for Speaker Diarization," arXiv:1911.01255, 2023.
[3] Zhang, J. et al., "PEGASUS: Pre-training with Extracted Gap Sentences for Abstractive Summarization," ICML, 2020.
[4] Rennard, V. et al., "Abstractive Meeting Summarization: A Survey," TACL, 2023.
[5] Kachhoria, R. et al., "Minutes of Meeting Generation for Online Meetings using NLP," IEEE ESCI, 2024.
[6] Lewis, M. et al., "BART: Denoising Sequence-to-Sequence Pretraining," ACL, 2020.
[7] Beltagy, I. et al., "Longformer: The Long-Document Transformer," arXiv, 2020.
[8] Devlin, J. et al., "BERT: Deep Bidirectional Transformers for Language Understanding," NAACL-HLT, 2019.
[9] Jurafsky, D. & Martin, J. H., Speech and Language Processing, Pearson, 2023.
[10] MongoDB Inc., "MongoDB Atlas Cloud Database Platform," 2024.
[11] Savitribai Phule Pune University — Project Evaluation Guidelines, Faculty of IT, 2024