IntelliAudio: AI Powered Understanding and Navigation of Long-Form Speech Content

1. Project Overview

Problem Statement

With the rapid growth of podcasts, lectures, and recorded discussions, long-form audio has become an important source of information. However, audio content is inherently sequential, making it difficult for users to quickly locate specific information without listening to the entire recording. Unlike text, audio cannot be easily skimmed or searched, which reduces accessibility and productivity for learners, researchers, and general listeners.

Objective of the Project

The goal of this project is to develop an AI-powered system capable of transforming long-form speech into structured and searchable information. The system performs multiple tasks including speech transcription, topic detection, summarization, keyword extraction, and sentiment analysis, along with an interactive interface for content navigation.

The specific objectives are:

- Convert speech into readable text using speech recognition
- Identify topic boundaries within conversations
- Generate concise summaries of discussions
- Extract meaningful keywords
- Analyze sentiment of different segments
- Provide an interactive visualization interface

Significance and Real-World Applications

The proposed system has practical applications in several domains:

- **Education:** Enables students to quickly access specific lecture topics

- **Accessibility:** Helps hearing-impaired users understand spoken content
- **Media and Research:** Assists analysts in reviewing interviews and discussions efficiently
- **Content Consumption:** Allows users to navigate podcasts without listening completely

---

1. 3. System Architecture

2. The system follows a multi-stage pipeline to convert raw audio into structured and interactive information. Each module performs a specific task and passes its output to the next stage, forming an end-to-end audio understanding workflow.
3. The stages of the system are:
4. Audio Input
   The user provides a long-form audio file such as a podcast or lecture recording.

5. Audio Preprocessing
   The audio is standardized by converting it to WAV format, resampling to 16 kHz mono channel, and reducing noise. Long recordings are divided into smaller chunks to enable efficient processing.

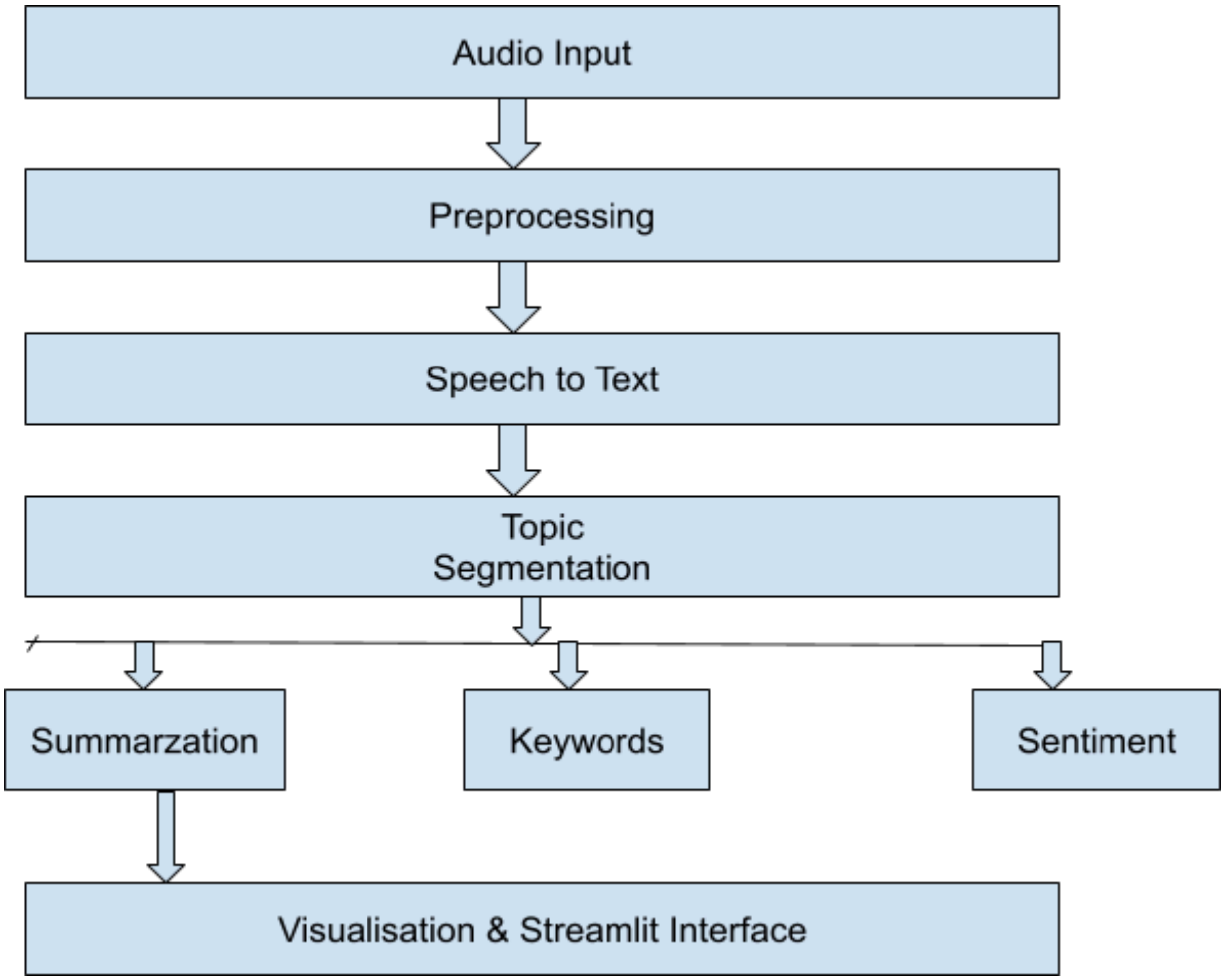
6. Speech-to-Text Conversion
   The processed audio segments are transcribed into text using a speech recognition model. Timestamps are preserved for synchronization with the original audio.

7. Topic Segmentation
   The transcript is analyzed to identify boundaries where the conversation shifts to a new topic. Semantic similarity between consecutive segments is used to detect topic transitions.

8. Summarization and Keyword Extraction
   Each topic segment is summarized and important keywords are extracted to represent the main discussion points.

9. Sentiment Analysis
   The emotional tone of each segment is analyzed to determine whether the conversation is positive, negative, or neutral.

10. Visualization and User Interface
    All processed information is presented in an interactive interface where users can read transcripts, navigate between topics, and explore keywords.



- **4. Tools and Libraries Used**
- **Audio Processing**

- **LibROSA**
  LibROSA was used for loading audio files, resampling them to a uniform sampling rate, and preparing them for speech recognition. It ensures that audio signals fall within the frequency range suitable for human speech processing.
- **PyDub**
  PyDub was used for audio format conversion and splitting long recordings into smaller chunks. This helps process long audio files efficiently without memory overflow during transcription.
- ---
- **Speech-to-Text**
- **OpenAI Whisper**
  Whisper was used for speech recognition due to its robustness to accents, pauses, and background noise. It provides timestamped transcripts, enabling synchronization between text and original audio segments.
- ---
- **Natural Language Processing**
- **Sentence-BERT**
  Sentence-BERT embeddings were used to measure semantic similarity between consecutive transcript segments. This helps detect topic boundaries more accurately compared to keyword-based approaches.
- **TF-IDF**
  TF-IDF was used for keyword extraction by identifying words that are important within a segment but less frequent across the entire transcript.
- **VADER Sentiment Analyzer**
  VADER was used to determine the sentiment polarity (positive, negative, neutral) of each topic segment. It is efficient for conversational text and does not require heavy computational resources.
- ---
- **Visualization and Interface**

- **Matplotlib / WordCloud**
  Used to generate visual keyword representations to help users quickly understand major discussion topics.
- **Streamlit**
  Streamlit was used to develop an interactive user interface that allows users to upload audio, view transcripts, navigate topics, and explore analysis results in real time.
    - # 5. Implementation Details
    - **Speech Transcription**
- The input audio is first divided into smaller chunks to handle long recordings efficiently. Each chunk is passed to the Whisper speech recognition model to generate text along with timestamps. The individual transcripts are then merged in chronological order to form the complete transcript of the audio.
    - ---
    - **Topic Segmentation**
- After transcription, the text is split into smaller sentences or segments. Sentence embeddings are generated using Sentence-BERT to capture semantic meaning. Cosine similarity is computed between consecutive segments, and a significant drop in similarity indicates a topic change. These boundary points are used to divide the transcript into meaningful discussion sections.
    - ---
    - **Summarization**
- Each topic segment is summarized by identifying important sentences based on relevance within the segment. The goal is to provide a concise representation of the discussion while preserving the core meaning of the conversation.
    - ---
    - **Keyword Extraction**
- TF-IDF scoring is applied to each segment to identify words that are highly relevant within that topic but less frequent in the entire transcript. These keywords represent the primary discussion points of the segment.
    - ---

- **Sentiment Analysis**

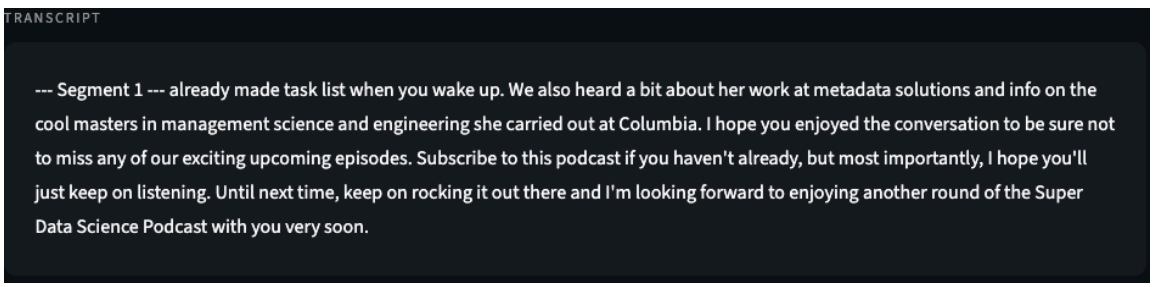
- Each segmented topic is analyzed using the VADER sentiment analyzer. The polarity score determines whether the segment conveys positive, negative, or neutral sentiment, enabling emotional understanding of the conversation flow.
  - ---
  - **Interactive Visualization**
- All generated outputs including transcript, topic segments, summaries, sentiment labels, and keywords are displayed in a Streamlit interface. The interface allows users to navigate between topics and explore the content efficiently.
  - ---

6. Results and Outputs

The developed system successfully converts long-form podcast audio into structured and interactive information. The output is presented through a user-friendly interface where users can navigate discussion segments, understand sentiment, and identify important keywords without listening to the entire recording.

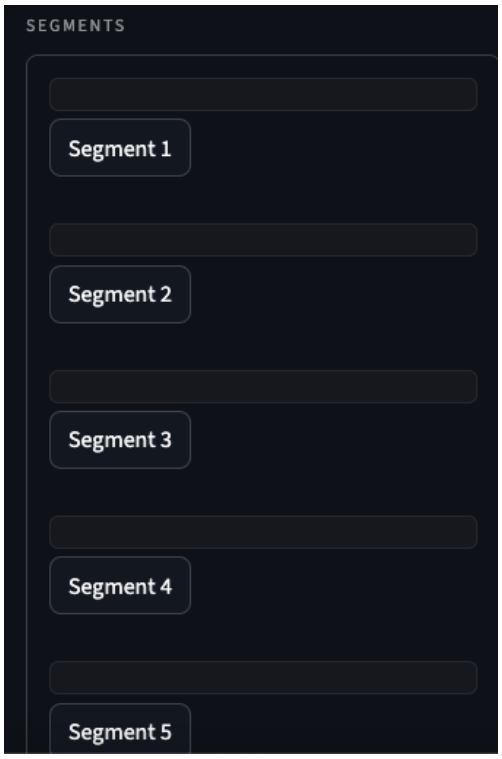
6.1 Generated Transcript

The speech recognition module converts spoken audio into readable text while preserving the sequence of the conversation. This allows users to understand the content of the recording without manually listening to the entire audio.



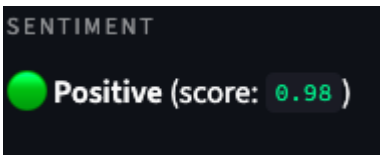
6.2 Topic Segmentation

The system automatically divides the transcript into multiple meaningful segments based on semantic similarity between consecutive sentences. Each segment represents a shift in discussion topic and can be individually



6.3 Sentiment Analysis

Each identified segment is analyzed to determine the emotional tone of the discussion. The system assigns a sentiment score indicating whether the conversation is positive, negative, or neutral. This helps users quickly understand the context of the discussion.

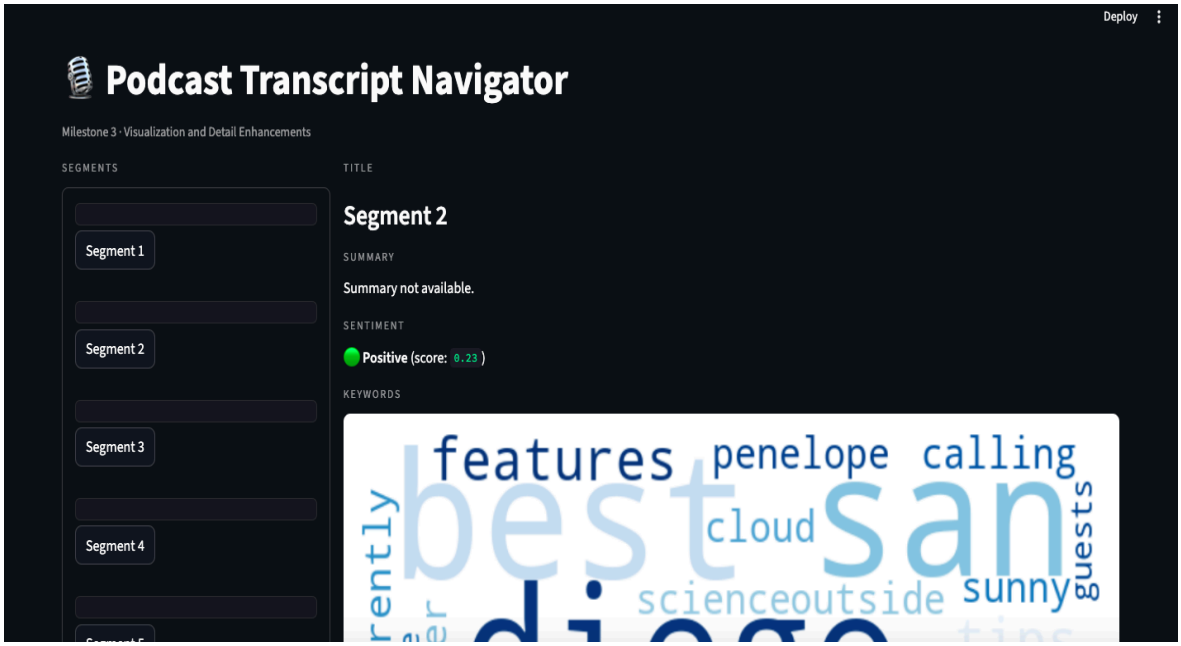


6.4 Keyword Visualization

Important words from each segment are extracted and displayed using a word cloud visualization. The size of each word indicates its importance within the discussion, allowing users to quickly identify major topics.

6.5 Interactive User Interface

The Streamlit interface allows users to upload audio, navigate between segments, view summaries, sentiment scores, and keywords in real time. This interactive environment makes long-form audio content easy to explore and understand.



7. Testing and Feedback

The system was tested on multiple podcast recordings to evaluate transcription quality, topic detection accuracy, and overall usability. During testing, several issues were identified and improvements were implemented to enhance performance and reliability.

Podcast Scenario	Issue Identified	Corrective Action Taken
Fast Speech Segments	Missing or incorrect words in transcript	Reduced chunk duration for better recognition
Background Noise	Unwanted Words Detected	Applied noise reduction preprocessing

Long Discussion Without pause	Incorrect Topic Boundaries	Adjusted Similarity Threshold in Segmentation
Intro Music Section	Non Speech Content Transcribed	Trimmed silent/non-speech portions
Mixed Discussion Topic	Overlapping Segment Detection	Used semantic similarity instead of keyword matching

User Feedback

Users were able to easily navigate podcast content without listening to the entire recording. The segment-based navigation significantly reduced time required to locate relevant information. The keyword visualization helped users quickly understand the main topic of discussion. Minor improvements were suggested in summary clarity for some segments.

8. Limitations

Although the system performs effectively for structured podcast analysis, certain limitations remain:

- Transcription accuracy may decrease when speakers talk very fast or overlap

- Background noise and music can affect speech recognition quality

- Topic segmentation may struggle when topic transitions are gradual
- Sentiment analysis may not correctly interpret sarcasm or complex emotions
- The system performance depends on the clarity and quality of the input audio

---

9. Future Work

Paste this:

9. Future Work

The system can be further improved by implementing advanced features:

- Speaker identification (speaker diarization) to distinguish multiple speakers
- Real-time audio transcription and analysis
- Integration of advanced language models for more accurate summarization
- Multilingual support for non-English audio content
- Search functionality to directly find specific information within transcripts
- Cloud deployment for public accessibility