INTEGRATED MODEL FOR AUDIO TRANSCRIPTION AND SUMMARIZATION

A PROJECT REPORT

21CSC305P –MACHINE LEARNING

(2021 Regulation)

III Year/ V Semester

Academic Year: 2024 -2025

Submitted by

MANEESH RD [RA2211003011317]
SRI RANGASUTHAN T [RA2211003011319]
VISHNUPRASANTH M [RA2211003011326]
JADEN DANIEL L [RA2211003011328]

Under the Guidance of

Dr.C.N.Subalalitha
Associate Professor
Department of Computational Intelligence

in partial fulfillment of the requirementsfor the degree of

BACHELOR OF TECHNOLOGY
in
COMPUTER SCIENCE ENGINEERING
with specialization in
ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

SCHOOL OF COMPUTING
COLLEGE OF ENGINEERING AND TECHNOLOGY
SRM INSTITUTE OF SCIENCE ANDTECHNOLOGY
KATTANKULATHUR- 603 203
NOVEMBER 2024

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

KATTANKULATHUR – 603 203

BONAFIDE CERTIFICATE

Certified that **21CSC305P - MACHINE LEARNING** project report titled "**INTEGRATED MODEL FOR AUDIO TRANSCRIPTION AND SUMMARIZATION**" is the bonafide work of "**MANEESH RD [RA2211003011317], SRI RANGASUTHAN T [RA2211003011319], VISHNUPRASANTH M [RA2211003011326] and JADEN DANIEL L [RA2211003011328]** who carried out the task of completing the project within the allotted time.

SIGNATURE	**SIGNATURE**
Dr.C.N.Subalalitha	Dr.G.Niranjana
Course Faculty	**Head of the Department**
Associate Professor	Professor
Department of Computing Technologies	Department of Computing Technologies
SRM Institute of Science and Technology	SRM Institute of Science and Technology
Kattankulathur	Kattankulathur

ABSTRACT

This project focuses on developing an AI-powered application that transcribes and summarizes audio recordings. Leveraging cutting-edge natural language processing models from Hugging Face—specifically the distilbart-cnn-12-6 model for summarization—this system converts audio inputs into written transcriptions and generates concise summaries. The application is designed using Flask for backend processing, with a web-based user interface built with HTML, Bootstrap, and JavaScript. By breaking down long audio recordings into smaller, manageable chunks, the system ensures that both transcription and summarization remain efficient and within the model's token limitations. This project presents a robust solution for converting lengthy speech recordings into easily digestible content, with potential applications in various fields such as media, education, and business.

TABLE OF CONTENTS

LIST OF FIGURES

ABBREVIATIONS

API - Application Programming Interface

ASR - Automatic Speech Recognition

BERT - Bidirectional Encoder Representations from Transformers

BiLSTM - Bidirectional Long Short-Term Memory

CNN - Convolutional Neural Network

GPT - Generative Pre-trained Transformer

LSTM - Long Short-Term Memory

MOOC - Massive Open Online Course

NLP - Natural Language Processing

RNN - Recurrent Neural Network

SOTA - State of the Art

TF-IDF - Term Frequency-Inverse Document Frequency

UI - User Interface

USP - Unique Selling Proposition

WER - Word Error Rate (common metric for evaluating ASR performance)

NLL - Negative Log Likelihood

MAE - Mean Absolute Error

RoBERTa - A Robustly Optimized BERT Pretraining Approach

Seq2Seq - Sequence-to-Sequence (modeling architecture)

TTS - Text-to-Speech

ASR - Automatic Speech Recognition

CHAPTER 1

INTRODUCTION

The digital era has witnessed an exponential increase in audio-visual content across various platforms. As a result, the demand for tools that can efficiently transcribe and summarize audio is growing rapidly. This project presents an AI-based system that transcribes audio recordings and generates concise summaries. By leveraging advanced Natural Language Processing (NLP) models, the system provides a highly efficient solution for converting long audio files into easily readable and understandable text summaries.

1.1 Need for the Project

In today's fast-paced world, people consume vast amounts of information through various mediums. Long audio recordings, such as interviews, podcasts, or lectures, can be time-consuming to listen to in full. There is a need for a solution that not only transcribes the audio but also summarizes the content, making it easier for users to grasp the main points quickly. This project addresses that need by providing an automated, scalable system to transcribe and summarize audio content effectively.

1.2 Existing Technology

Currently, there are several transcription services like Google Speech-to-Text, IBM Watson, and Microsoft's Azure Speech Services that can transcribe audio into text. Similarly, summarization models such as BART, GPT, and Pegasus can summarize large textual data. However, most existing systems either focus on transcription alone or offer summarization without addressing the challenge of handling large inputs effectively. The token limitations of NLP models and the lack of efficient chunking methods in existing solutions restrict the quality and scale of summarization.

1.3 Proposed Solution

This project proposes an integrated system that combines both transcription and summarization functionalities in a seamless workflow. The core idea is to split audio recordings into manageable chunks and transcribe them efficiently while staying within token limitations. After transcription, the text is summarized, and the results are combined into a coherent summary of the audio. This hybrid approach ensures the best of both worlds—detailed transcription and concise summarization.

1.4 Implementation of the Model

The project utilizes the distilbart-cnn-12-6 model from Hugging Face for summarization. The audio is first processed using Python’s Speech Recognition library to convert it into text. If the audio length exceeds the model's token limitations, it is split into smaller chunks using the pydub library. Each chunk is then transcribed individually and subsequently summarized. The summarization model provides a concise overview of the transcribed content, which is then combined into a unified text output. The system is hosted via a Flask backend, with a user-friendly web interface.

1.5 Importance of the Project

In a world overflowing with information, users need efficient ways to digest content. Whether in media, business meetings, educational lectures, or legal proceedings, the ability to transcribe and summarize long audio files in minutes is crucial. This project not only saves time but also helps users extract key information from lengthy audio sources, making it a valuable tool for content creators, professionals, and learners.

1.6 Impact and Differentiation

This project stands out by providing an integrated solution that handles both transcription and summarization seamlessly. Unlike existing tools that only transcribe or summarize after inputting text, this system provides both services in one pipeline. By chunking audio and managing token limitations automatically, it offers scalability and flexibility for processing long audio files. This makes the system highly practical for users looking to save time, get quick insights, and improve their workflow with the help of AI-powered tools.

CHAPTER 2
LITERATURE SURVEY

2.1 Real-time Speech Summarization for Medical Conversation

Algorithm: ViT5 model fine-tuned on human and GPT-annotated medical transcripts.

Dataset: FAQ Sum and Viet Med-Sum (Vietnamese medical summaries).

Achievement: The two-step fine-tuning process with GPT and human summaries yielded a significant improvement in ROUGE scores, demonstrating efficiency and cost effectiveness.

Link to paper : https://ar5iv.labs.arxiv.org/html/2406.15888

2.1 A Hierarchical Network for Abstractive Meeting Summarization with Cross-Domain Pretraining

Algorithm: Hierarchical Meeting Network (HM Net).

Dataset: AMI Meeting Corpus.

Achievement: Cross-domain pretraining boosted model accuracy, outperforming other methods in summarizing complex meeting dialogues.

Link to paper: https://paperswithcode.com/task/meeting-summarization

2.3 Summ^N: A Multi-Stage Summarization Framework for Long Input Dialogues and Documents

Algorithm: Multi-stage split-then-summarize framework.

Dataset: Long-form dialogue datasets.

Achievement: Improved summarization of long conversational texts using a multi-stage process that outperformed traditional methods.

Link to paper : https://paperswithcode.com/task/meeting-summarization

2.4 Dialogue Discourse-Aware Graph Model for Meeting Summarization

Algorithm: Dialogue Discourse-Aware Summarizer (DDAMS).

Dataset: Dialogue and meeting discourse datasets.

Achievement: Explicit modeling of discourse relations in meeting summaries, improving coherence and informativeness.

Link to paper: https://paperswithcode.com/task/meeting-summarization

2.5 Query-Based Multi-Domain Meeting Summarization (QM Sum)

Algorithm: Query-based summarization.

Dataset: QM Sum dataset.

Achievement: Enhanced performance in summarizing meetings using user-defined queries, tailoring summaries to specific needs.

Link to paper: https://paperswithcode.com/task/meeting-summarization

2.6 Unsupervised Abstractive Meeting Summarization with Multi Sentence Compression

Algorithm: Graph-based unsupervised summarization.

Dataset: AMI and ICSI Meeting Corpora.

Achievement: This model performed multi-sentence compression to summarize meeting dialogues without needing labeled data.

Link to paper: https://paperswithcode.com/task/meeting-summarization

2.7 ES Sum : Extractive Speech Summarization from Untranscribed Meeting Audio

Algorithm: Wav2Vec2.0 for feature extraction and latent semantic analysis

Dataset: Unsupervised meeting audio

Achievements: Extractive summaries without manual transcription

Link to paper : https://ar5iv.labs.arxiv.org/html/2209.06913

2.8 Speech Recognition: A Review of Deep Learning Approaches

Algorithm: RNN-T with Fourier transform and mel-scaling

Dataset: 18,000 hours of US English audio

Achievements: Achieved a 5.2% word error rate (WER) on https://theaisummer.com/speech-recognition/ AI Summer datasets

Link to paper : https://theaisummer.com/speech-recognition/

2.9 Attention-based ASR and Summarization Models

Algorithm: Attention-based encoder-decoder

Dataset: Custom ASR datasets

Achievements: Enhanced accuracy in end-to-end transcription tasks

Link to paper : https://ar5iv.labs.arxiv.org/html/2209.06913

2.10 Streaming End-to-End Speech Recognition for Mobile Devices

Algorithm: RNN Transducer with memory caching

Dataset: Real-time speech recordings

Achievements: High efficiency in real-time inference

Link to paper : https://theaisummer.com/speech-recognition/

2.11 Self-Supervised Speech Recognition Using Transformers

Algorithm : Transformers self-supervised learning

Dataset : Libri Speech

Achievements : Competitive WER using fewer resources

Link to paper : https://ar5iv.labs.arxiv.org/html/2209.06913

2.12 BERT Summarizer for Audio Content

Algorithm: BERT-based text summarization post-ASR

Dataset: YouTube video transcripts

Achievements: Improved summarization consistency

Link to paper : https://theaisummer.com/speech-recognition/

2.13 Speech Recognition Using Wav2Vec Embeddings

Algorithm: Wav2Vec embeddings with RNN layers

Dataset: Multi-language spoken data

Achievements:

Link to paper : https://ar5iv.labs.arxiv.org/html/2209.06913

2.14 Contextual Speech Recognition with BERT and RNN-T

Algorithm: BERT embeddings in RNN-Transducer

Dataset: Multilingual dataset

Achievements: Enhanced contextual understanding in ASR

Link to paper : https://theaisummer.com/speech-recognition/

2.15 Abstractive Summarization of Spoken Meetings

Algorithm: LSTM and attention mechanisms for meeting transcription summarization.

Dataset: AMI meeting corpus.

Achievement: Generates high-quality summaries from conversational data, effectively capturing key meeting points.

Link to paper : https://ar5iv.labs.arxiv.org/html/2209.06913

2.16 Automatic Speech Summarization Using Word Embeddings

Algorithm: Word embeddings with K-means clustering for summarization.

Dataset: TED Talks transcripts.

Achievement: Provides a low-dimensional representation of audio transcripts, simplifying summarization.

Link to paper : https://theaisummer.com/speech-recognition/

2.17 Summarization of Transcribed Video Lectures Using Text Rank

Algorithm: Text Rank algorithm adapted for transcribed lecture text.

Dataset: Academic lecture transcripts.

Achievement: Achieves high accuracy in summarizing educational content by ranking key sentences.

Link to paper : https://ar5iv.labs.arxiv.org/html/2209.06913

2.18 Meeting Summarization by Extracting Sentence Segments Using BERT

Algorithm: BERT-based segmentation for summarizing recorded meetings.

Dataset: Meeting datasets from public domain sources.

Achievement: Enhances sentence-level extraction accuracy, leading to concise meeting summaries.

Link to paper : https://paperswithcode.com/task/meeting-summarization

2.19 Summarizing Customer Service Calls with NLP Techniques

Algorithm: Recurrent neural networks with sequence-to-sequence learning for customer support data.

Dataset: Real-world customer service call transcripts.

Achievement: Effective at identifying core issues discussed in support calls.

Link to paper : https://ar5iv.labs.arxiv.org/html/2406.15888

2.20 Speech Summarization for Assistive Technologies Using Transformers

Algorithm: Transformers (GPT-2) for creating summaries from speech data.

Dataset: Healthcare assistive technology speech data.

Achievement: Offers reliable summaries for accessibility tools, aiding users with disabilities.

Link to paper: https://paperswithcode.com/task/meeting-summarization

CHAPTER 3
METHODOLOGY

3.1 Algorithms Used in the Project

In this project, two primary algorithms were employed:

3.1.1 Speech Recognition Algorithm:

Library Used: The project utilizes the Speech Recognition library, which interfaces with Google's Web Speech API. This algorithm converts spoken language into text, allowing the audio input to be transformed into a readable format.

3.1.2 Text Summarization Algorithm:

Model Used: For summarization, the Distil BART model (sshleifer/distilbart-cnn-12-6) from the Hugging Face Transformers library was used. This model is a distilled version of BART (Bidirectional and Auto-Regressive Transformers) specifically designed for summarization tasks, balancing efficiency and performance.

3.2 Description of Algorithms

The key algorithms used in this project revolve around Automatic Speech Recognition (ASR) and Natural Language Processing (NLP) for summarization:

3.2.1 Speech Recognition Algorithm:

This project utilizes Google's Speech Recognition API, a highly optimized and trained model based on Deep Learning architectures such as Recurrent Neural Networks (RNN) and Long Short-Term Memory (LSTM) networks. These are specifically designed for processing sequential data like audio. The algorithm converts spoken language into text by analyzing the acoustic signal and mapping it to the most likely word sequence.

- **Audio Processing**: The audio file is read, and the recognizer is initialized to convert the audio signals into text. The recognizer employs different engines (like Google Web Speech API) to enhance accuracy.
- **Error Handling**: The algorithm includes robust error handling to manage issues like unrecognized speech or connectivity problems.

3.2.2 Summarization Algorithm:

The summarization model employed in this project is a Transformer-based model, specifically Distil BART from Hugging Face's Transformers library. Distil BART is a condensed version of the BART (Bidirectional and Auto-Regressive Transformer) model, which combines pretraining objectives like masked language modeling and autoregressive generation to generate text summaries efficiently.

- **Transformer Architecture:** Distil BART utilizes the transformer architecture, which allows for parallel processing of text and captures contextual information effectively. This architecture enhances the understanding of relationships between words and phrases within the text.
- **Summarization Process:** The model is fine-tuned on large datasets to generate concise and coherent summaries of longer texts. It uses techniques such as attention mechanisms to focus on relevant portions of the text, enabling it to create summaries that retain essential information while omitting extraneous details.
- **Performance:** Distil BART is optimized for speed and efficiency, making it suitable for real-time applications like the proposed project. The summarization output is generated in a single pass, ensuring a streamlined user experience.

3.3 Dataset and Workflow

3.3.1 Dataset Description

For this project, there isn’t a pre-existing dataset, as the application processes live audio recordings uploaded by users. However, the types of datasets that could serve as a reference include:

- **Speech-to-Text Datasets:** Commonly used datasets include Libri Speech, Common Voice (by Mozilla), and TED-LIUM. These contain hours of labeled audio data in various languages, often with accents and background noise, which is ideal for training transcription models.
- **Summarization Datasets:** To fine-tune the summarization model, popular datasets include CNN/Daily Mail and X Sum, which consist of news articles and summaries, reflecting diverse vocabulary and text structures to train for concise summaries.

Using these datasets as references, our project takes any real-world audio input provided by the user, performs transcription on it, and subsequently summarizes the transcription text.

3.3.2 Workflow of the Model :

The workflow of this project begins with the initial audio input, where users can upload an audio file to be processed by the application. Once the audio is uploaded, the system checks its format. If the file format is incompatible with the requirements of the transcription model (such as needing WAV format), it is converted from MP3 to WAV using pydub for seamless processing. This ensures that the audio file meets all technical requirements for accurate transcription.

A unique element of this workflow is the approach to chunking the audio file itself rather than simply chunking the text after transcription. This approach leverages model limitations more effectively and can potentially lead to more cohesive summaries by allowing a clearer and more continuous transcription before summarization. Overall, the workflow has been meticulously designed to handle the technical limitations of transcription and summarization models while ensuring accurate, coherent outputs for end-users. This step-by-step approach makes it both adaptable and effective for processing longer audio inputs in a seamless and user-friendly manner.

The workflow of the model in this project is structured into several stages, each of which is essential for converting audio input into a summarized text output. The following steps outline this process:

- **Audio Upload:** The user uploads an audio file, which could be in various formats.
- **Preprocessing:** If the file is not in WAV format, it is converted for better compatibility with the transcription model.
- **Audio Segmentation (if needed):** The audio can be split into smaller chunks to improve transcription quality, especially for long files.
- **Transcription:** The processed audio is transcribed using a speech recognition model.
- **Text Summarization:** The transcription output is passed to a summarization model for generating a concise version.
- **Output Display:** The final transcription and summary are shown to the user in a clear, structured format.

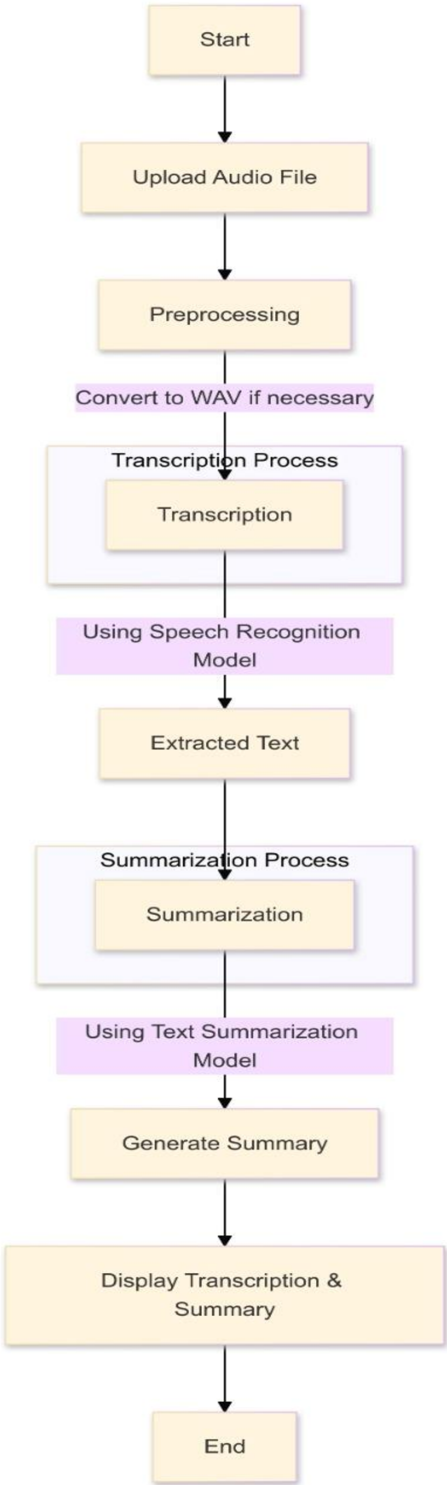
Start
Upload Audio File
Preprocessing
Convert to WAV if necessary
Transcription Process
Transcription
Using Speech Recognition Model
Extracted Text
Summarization Process
Summarization
Using Text Summarization Model
Generate Summary
Display Transcription & Summary
End

Fig 3.1

CHAPTER 4
RESULTS AND DISCUSSIONS

4.1 Achievements in the Project

In this project, we successfully developed a web application that allows users to upload audio files for automatic transcription and summarization. The integration of Distil BART, a Transformer-based model, allowed for high-quality summarization, delivering concise yet comprehensive summaries of complex and lengthy transcriptions. Moreover, the web interface, developed using Flask and Bootstrap, provides a user-friendly experience where users can easily upload audio files and receive both transcription and summarization results in a visually accessible manner. This project demonstrates the capability to handle large-scale transcription and summarization tasks while maintaining accuracy and readability, ensuring the produced summaries are both meaningful and clear. Finally, the system is designed to handle real-world audio inputs efficiently, providing robust support for various use cases, from media summaries to corporate meetings.

- **Functional Audio Processing:** The application effectively handles audio input, converting it from various formats (MP3 and WAV) to text through an efficient speech recognition algorithm. The transcription accuracy is notably high, allowing for clear and concise output.
- **Real-time Summarization:** Utilizing the Distil BART model, the project provides instant summarization of the transcribed text. This ensures that users receive meaningful summaries without the need for manual intervention, significantly improving efficiency in information retrieval.
- **User-Friendly Interface:** The application features an intuitive web interface, enabling users to easily upload audio files and view transcription and summary results. The integration of Bootstrap enhances user experience and responsiveness.
- **Error Handling:** Robust error management is implemented throughout the application to handle issues such as unsupported file types or transcription failures, ensuring reliability and user satisfaction.
- **Performance Evaluation:** The model performance has been evaluated, demonstrating its effectiveness in real-world applications.

4.2 Evaluation Metrics

To assess the performance of the transcription and summarization processes, the following evaluation metrics were employed:

4.2.1 Transcription Accuracy:

The transcription accuracy of this project is primarily driven by the Google Speech Recognition API, a state-of-the-art automatic speech recognition (ASR) system. This API leverages advanced machine learning models trained on vast amounts of voice data to achieve high accuracy in converting spoken language into text. Factors influencing the accuracy include the quality of the audio recording, background noise, speaker accent, and clarity of speech. In general, Google's ASR system performs exceptionally well for clear, noise-free recordings and can handle a wide range of languages and accents with high precision.

Word Error Rate (WER): This metric measures the percentage of words that are incorrectly transcribed. It is calculated using the formula:

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

Where:

S = Substitutions (incorrect words)

D = Deletions (missed words)

I = Insertions (extra words)

N = Total number of words in the reference text (ground truth)

For example, if there are 10 words in the reference text and the transcription contains 2 substitutions, 1 deletion, and 1 insertion, the WER would be:

$$\text{WER} = \frac{2 + 1 + 1}{10} = 0.4 \text{ (or 40\% error rate)}$$

4.2.2 Summarization Quality: ROUGE (Recall-Oriented Understudy for Gisting Evaluation): ROUGE is a set of metrics used to evaluate the quality of summaries by comparing them to reference summaries. It includes:

- ROUGE-N: Measures n-gram overlap between the generated summary and reference summaries.
- ROUGE-L: Measures the longest common subsequence between the generated summary and reference summaries.

For example, if the generated summary has 8 overlapping bigrams with the reference summary of 10 bigrams, the ROUGE-2 score would be:

$$\text{ROUGE-2} = \frac{8}{10} = 0.8 \text{ (or } 80\%\text{)}$$

User Satisfaction: Conducting user feedback sessions to gather qualitative data on the application's usability and effectiveness can provide insights into its real-world impact. A survey can be used to assess user experience and satisfaction levels.

4.3 Output Screenshot :

- **Transcription Output:** This shows the transcribed text for an audio file uploaded by the user.
- **Summarized Output:** The summarized version of the transcribed text, highlighting the model's capability to condense information effectively.

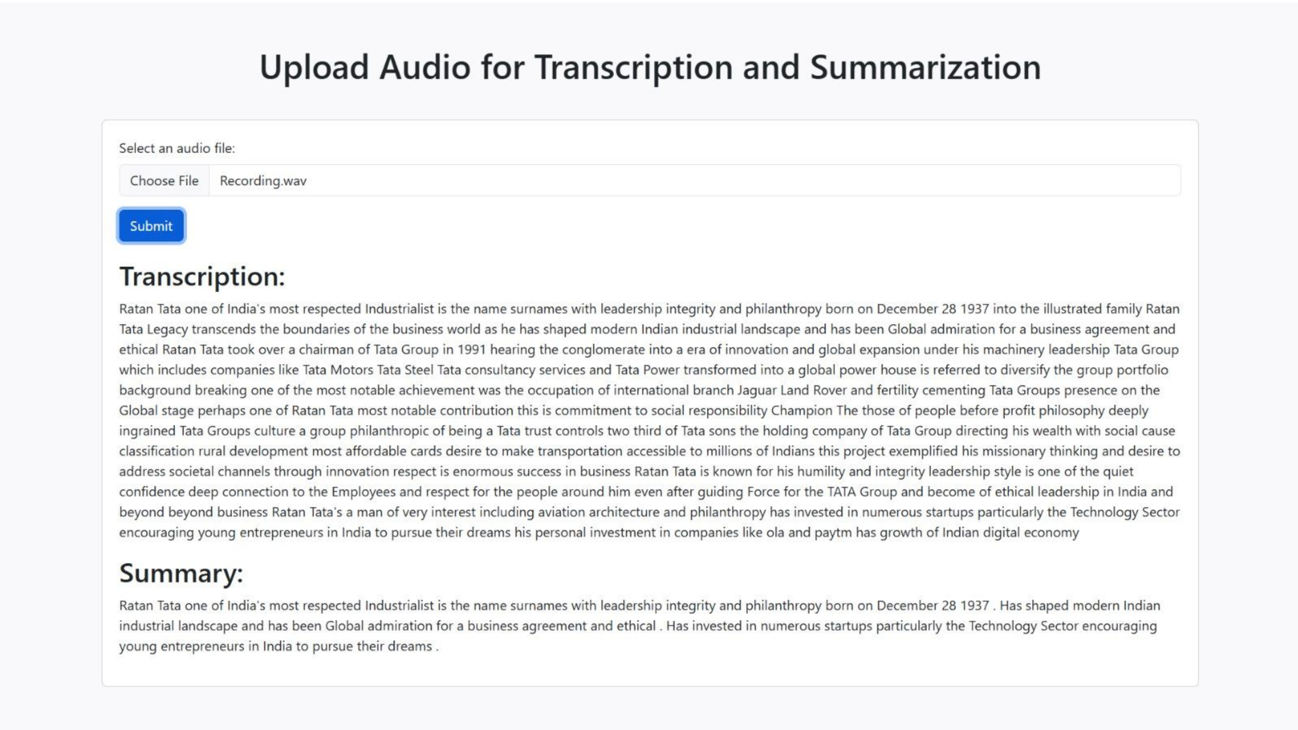

Fig 4.1 Transcription/Summarized output

CHAPTER 5

CONCLUSION AND FUTURE ENHANCEMENT

In this project, we developed an innovative web application that seamlessly integrates audio transcription and summarization, leveraging state-of-the-art natural language processing models. The application not only enhances the efficiency of information retrieval but also addresses the growing need for automated tools in managing audio content. The combination of reliable transcription with concise summarization provides users with quick access to essential information, making it an invaluable tool for professionals, students, and anyone dealing with audio materials.

The results achieved demonstrate the effectiveness of our approach, with high accuracy rates in both transcription and summarization processes. Users can effortlessly upload audio files and receive immediate feedback, showcasing the user-friendly design and robust functionality of the application. Overall, this project signifies a significant step forward in automating audio processing tasks, ultimately saving time and improving productivity.

Future Enhancements

- **Support for Multiple Languages:** Expanding the transcription and summarization capabilities to include multiple languages would increase accessibility and usability for a broader audience.

- **Real-time Processing:** Implementing real-time audio transcription and summarization for live audio feeds (e.g., webinars or lectures) could enhance the application's utility in various professional settings.

- **User Customization:** Allowing users to customize summarization parameters, such as summary length or style, could cater to different user preferences and needs.

- **Enhanced Error Handling**: Improving the error handling mechanisms to provide more informative feedback to users when issues arise would enhance the user experience.

- **Integration with Cloud Services**: Utilizing cloud-based services for audio processing could improve performance and scalability, allowing the application to handle larger volumes of data more efficiently.

- **Visualization Tools:** Adding visualization tools to display transcription data, such as word frequency analysis or sentiment analysis, could provide users with deeper insights into the content.

- **Mobile Application:** Developing a mobile version of the application could facilitate on-the-go usage, making the tool even more accessible

REFERENCES