SPEECH TO TEXT CONVERSION

A PROJECT REPORT
for
Introduction To AI (AI101B)
Session (2024-25)

Submitted by

<Riya Gaur>
(202410116100171)
<Riya>
(202410116100170)
<Priyanshi>
(202410116100153)
<Pallavi>
(202410116100139)

Submitted in partial fulfilment of the
Requirements for the Degree of

MASTER OF COMPUTER APPLICATION

Under the Supervision of
<Ms. Komal Salgotra>

Submitted to
DEPARTMENT OF COMPUTER APPLICATIONS
KIET Group of Institutions, Ghaziabad
Uttar Pradesh-201206
(MONTH 2025)

ABSTRACT

The "Speech-to-Text Converter using Python" is an intelligent and accessible solution that converts spoken audio into written text. This project utilizes Python libraries such as SpeechRecognition, pydub, and IPython to process audio files, convert them into a readable format, and display the results on-screen. The primary goal of this project is to demonstrate how audio processing, machine learning, and API-based services can be integrated to simplify human-computer interaction. It is especially useful in domains like accessibility technology, automation, education, and virtual assistance. The project offers an efficient, cost-effective, and user-friendly tool for converting speech to text, laying the foundation for more advanced applications in voice recognition systems.

The system takes an audio file as input, converts it (if needed) into a suitable format, and then extracts the spoken words using Google's Speech Recognition API. It is especially useful for automating transcription tasks and can be extended for real-world uses in accessibility, education, and AI applications. This project demonstrates the integration of voice recognition and Python programming, making it ideal for students and beginners interested in real-world applications of AI.

INTRODUCTION

In the rapidly evolving landscape of modern technology, voice-enabled applications have become an essential part of everyday life. From virtual assistants like Amazon Alexa, Apple's Siri, and Google Assistant to automated customer support systems, speech recognition technology is driving a major shift in how humans interact with machines. The ability to convert spoken language into text with high accuracy has opened up countless possibilities across various industries including education, healthcare, accessibility services, smart home automation, and more.

This project, titled “Speech to Text Converter using Python,” is a simple yet practical implementation of a fundamental speech recognition system. It is designed to allow users to upload an audio file (in formats like MP3 or WAV), and then automatically transcribe the spoken content into written text using Python libraries such as SpeechRecognition and pydub. This process mimics the real-world functionality of voice assistants by making machines capable of listening to and understanding human speech.

The primary goal of this project is to demonstrate how user-recorded or pre-recorded audio can be processed and interpreted using Python. It also aims to make voice-based interaction more accessible by providing a tool that can be especially useful for those who are unable to type or have physical disabilities. Moreover, it highlights how Python, with its powerful libraries and clean syntax, can be leveraged to build efficient and beginner-friendly real-world applications.

By focusing on audio processing, format conversion, speech recognition, and error handling, this project not only serves an educational purpose but also reflects a practical use case of machine learning and artificial intelligence. Overall, the project introduces users to the basics of voice technology and serves as a stepping stone toward developing more complex AI-driven applications in the future.

SCOPE OF PROJECT

This project is not just limited to simple audio-to-text conversion but can also be used as a stepping stone for future enhancements. The following outlines the broader scope:

1. Basic Functionality: Convert .mp3 or .wav audio files to text.
2. Platform Independent: Can run on any system with Python support (Google Colab used here).
3. Educational Purpose: Ideal for students to understand API integration and audio processing.
4. Future Expansion: Can be upgraded for live speech recognition using a microphone, multilingual transcription, or even voice-command-based automation.
5. Real-World Usage: Acts as a mini-model for bigger systems like automatic transcription services, smart assistants, or AI-based dictation tools.

The scope of this project is both educational and practical. It is designed to demonstrate the real-world application of Python in speech processing.

- Supports basic audio file formats like .mp3 and .wav.
- Converts spoken content into editable and copyable text.
- Offers a foundation for voice assistants and dictation tools.
- Can be further developed into real-time transcription systems.
- Can be extended for multilingual speech recognition and live microphone input.

SIGNIFICANCE OF PROJECT

This project holds significant importance in the field of accessibility, AI, and human-computer interaction. Here's why:

1. Improves accessibility for hearing-impaired users or those with mobility issues.
2. Saves time and effort by reducing the need to manually type lengthy audio content.
3. Encourages the use of voice-based input systems in digital platforms.
4. Introduces students and developers to speech recognition, audio processing, and API usage in Python.
5. Can be extended into more intelligent systems like voice bots, virtual assistants, and automatic note-takers.

The significance of the project lies in how it simplifies a complex process like speech recognition using easy-to-understand Python code. The use of open-source libraries allows developers and students to explore real-world programming concepts in an accessible way.

1. Promotes Accessibility – Can help people who cannot type (e.g., physically disabled).
2. Reduces Manual Work – Great for transcribing interviews, lectures, or meetings.
3. Encourages Learning – Teaches how to work with external APIs and process media files.
4. Builds Foundation – Introduces beginners to concepts of AI, NLP (Natural Language Processing), and automation.

METHODOLOGY OF THE PROJECT

Step 1: Upload Audio

User uploads an .mp3 or .wav file using the Google Colab file uploader interface.

Step 2: Playback & Verification

The uploaded file is played using IPython's audio display so the user can confirm it's the correct file.

Step 3: Audio Conversion

If the uploaded file is in .mp3 format, it is converted to .wav using pydub, as the Google Speech API only supports WAV files.

```
sound = AudioSegment.from_mp3(audio_file)
sound.export("converted.wav", format="wav")
```

Step 4: Speech Recognition

Using Python's speech_recognition.Recognizer() object, the audio is transcribed into text.

```
with sr.AudioFile(wav_file) as source:
    audio_data = recognizer.record(source)
    text = recognizer.recognize_google(audio_data)
```

Step 5: Output Display

The recognized speech is printed as text in the output cell.

Step 6: Error Handling

The code is designed to catch:

UnknownValueError (if speech is unclear)

RequestError (if API connection fails)

Features of the Project

1. Upload and convert audio from .mp3 to .wav

2. Playback audio before processing
3. Uses Google's API for accurate speech recognition
4. Converts speech to clear, readable text
5. Error handling to manage bad audio or internet issues
6. Can be run in Google Colab — no installation needed
7. Beginner-friendly and customizable code

Real World Application

1. Transcription Tools – Used in education, media, and journalism for converting speech to notes.
2. Accessibility Software – Helps users with physical impairments who cannot type.
3. AI Voice Assistants – Forms the base logic of smart devices like Alexa, Siri, and Google Assistant.
4. Voice Command Interfaces – Smart homes and devices use similar technology.
5. Machine Learning and NLP – Can be used in training data for models in AI and voice analytics.

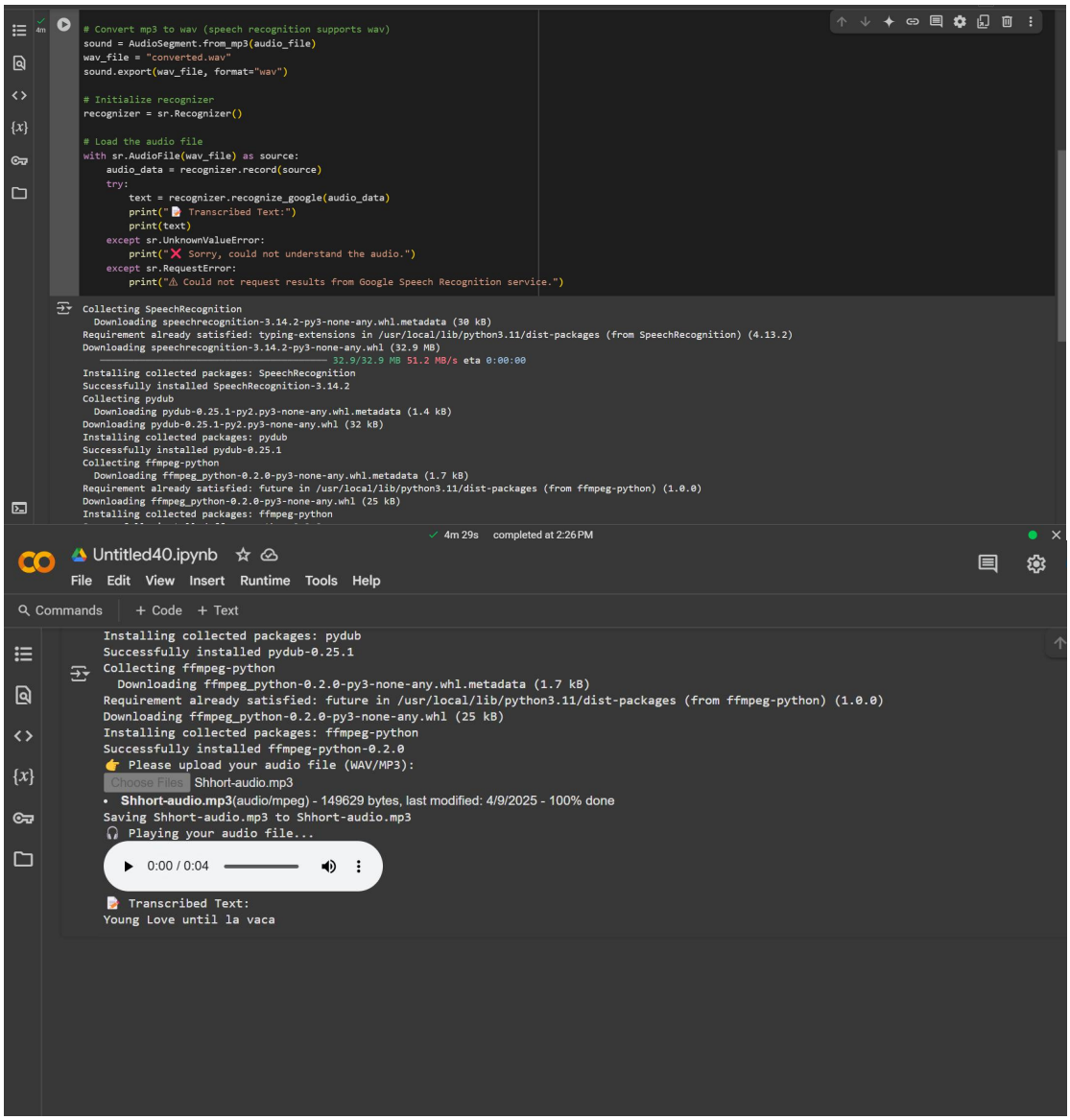
CODE

```
# Install required libraries
!pip install SpeechRecognition
```

```
!pip install pydub
!pip install ffmpeg-python            import speech_recognition as sr
from pydub import AudioSegment
from pydub.playback import play
import os
from google.colab import files    import IPython.display as ipd

# Upload audio
print("    Please upload your audio file (WAV/MP3):")
uploaded_audio = files.upload()

# Get filename
for file_name in uploaded_audio.keys():
    audio_file = file_name

# Play audio
print("    Playing your audio file...")
ipd.display(ipd.Audio(audio_file))      import speech_recognition as sr
import os
from pydub import AudioSegment

# Convert mp3 to wav (speech recognition supports wav)
sound = AudioSegment.from_mp3(audio_file)
wav_file = "converted.wav"
sound.export(wav_file, format="wav")

# Initialize recognizer
recognizer = sr.Recognizer()

# Load the audio file
with sr.AudioFile(wav_file) as source:
    audio_data = recognizer.record(source)
    try:
        text = recognizer.recognize_google(audio_data)
```

```
    print("    Transcribed Text:")
    print(text)
except sr.UnknownValueError:
    print("❌ Sorry, could not understand the audio.")
except sr.RequestError:
    print("⚠ Could not request results from Google Speech Recognition service.")
```

SCREENSHOT OF THE PROJECT


Untitled40.ipynb
File Edit View Insert Runtime Tools Help
Commands + Code + Text
Installing collected packages: pydub
Successfully installed pydub-0.25.1
Collecting ffmpeg-python
Downloading ffmpeg_python-0.2.0-py3-none-any.whl.metadata (1.7 kB)
Requirement already satisfied: future in /usr/local/lib/python3.11/dist-packages (from ffmpeg-python) (1.0.0)
Downloading ffmpeg_python-0.2.0-py3-none-any.whl (25 kB)
Installing collected packages: ffmpeg-python
Successfully installed ffmpeg-python-0.2.0
Please upload your audio file (WAV/MP3):
Choose Files Shhort-audio.mp3
Shhort-audio.mp3(audio/mpeg) - 149629 bytes, last modified: 4/9/2025 - 100% done
Saving Shhort-audio.mp3 to Shhort-audio.mp3
Playing your audio file...
0:00 / 0:04
Transcribed Text:
Young Love until la vaca

CONCLUSION

This project successfully demonstrates a basic and useful application of speech recognition using Python. It allows users to upload and process audio files into readable text, all while using open-source libraries and simple code logic. The knowledge gained here can be applied in more advanced projects like voice bots, live transcription apps, and AI assistants.

This not only showcases the power of Python in handling real-world problems but also encourages further learning in areas like AI, NLP, and human-computer interaction.

The Speech-to-Text Converter project is a practical demonstration of how voice and machine learning technologies can improve user interaction with digital systems. It shows how simple tools can become powerful when combined thoughtfully using Python. This project provides a solid base for creating advanced voice recognition applications and highlights the growing importance of voice interfaces in the digital world.