

# Speech To Text & Text to Speech

**Group Name:** py coders

**Group Members:**

| <b>First name</b> | <b>Last Name</b> | <b>Student number</b> |
|-------------------|------------------|-----------------------|
| Bhanu Prakash     | Mahadevuni       | C0850515              |
| Deeksha           | Naikap           | C0835440              |
| Pramod Reddy      | Gurralla         | C0850493              |
| Sai Varun         | Kollipara        | C0828403              |

Lambton College - AIMT Program  
2022W AML 1214 1 – Python Programming  
Simrandeep Kaur

**Submission date:** 21 April 2022

## Table of Contents

| <b>Title</b>                       | <b>Page No</b> |
|------------------------------------|----------------|
| <b>Abstract</b>                    | <b>3</b>       |
| <b>Introduction</b>                | <b>4</b>       |
| <b>Packages and Libraries</b>      | <b>5</b>       |
| <b>Methods</b>                     | <b>6</b>       |
| <b>Results</b>                     | <b>9</b>       |
| <b>Conclusions and Future Work</b> | <b>14</b>      |
| <b>References</b>                  | <b>15</b>      |

**Abstract:**

In the modern world, speech is the most potent way of communication. But there are different languages, dialects, pronunciations, etc., in the world, which makes it difficult for a more significant number of people worldwide to understand each other's conversations or their texts. So, to communicate without any hindrance, speech recognition makes things easy to comprehend each other conversations by converting speech into the required languages. Nowadays, speech recognition is also implemented in various appliances like Alexa, Siri, Google Assistant, etc. So, in this report, we will present the speech-to-text or text-to-speech application and the libraries required for our application.

**Introduction:**

Text-to-Speech and Speech-to-text is an application that takes input either in speech or text and converts it to text or speech in the selected language based on the chosen module. We will provide our speech input for the speech-to-text module and get the text as output in the language selected over the streamlit window. We can give the text input for the text-to-speech module and get speech as output in the language chosen over the streamlit window. The Application is developed using Google cloud APIs, speech recognition, and streamlit for GUI. Streamlit is an open-source python app framework that helps in building web applications.

## Packages and Libraries:

There are multiple packages and libraries that are implemented for the project development. To create the module, we installed the following libraries.

- Speech Recognition: For performing speech recognition
  - pip install SpeechRecognition
- Py\_Audio: Provide python bindings for port audio
  - First method: pip install PyAudio
  - Second method:
    - pip install pipwin
    - pipwin install pyaudio
- pyttsx3: Text-to-Speech conversion library
  - pip install pyttsx3
- Streamlit Library : For GUI implementation. Can represent the code in the format of web application.
  - pip install streamlit
- Google cloud text to speech: Google cloud API to convert text to speech
  - pip install google-cloud-texttospeech
- Google cloud translate: Google cloud API to translate the text from one language to other language
  - pip install google-cloud-translate
- The API from the google clod should be loaded for the usage of the functions. The google APIs are free for usage and can be gathered easily.
  - Login to Google Cloud
  - Search for the text-to-speech and speech-to-text, now add those to your cloud account.
  - The API codes will be available in the json format.
  - Load the API to the code based on the requirement.

## Methods:

The initial proposal is to build 2 modules that can perform tasks like converting speech to text and vice versa. And later, another two modules are added: the translation module and GUI (Graphical User Interface) modules. We will look at all the modules and how they perform with real-time inputs and variables. All the modules are described below:

1. Speech-to-Text Module
2. Text-to-Speech Module
3. Translate Module
4. GUI Module

Some of the requirements must be met to implement the modules, and we will discuss them. Initially, some packages and libraries must be downloaded and imported for the primary working functions. Next, the APIs we are implementing have some codes generated based on the account/user. All the steps and requirements are discussed in the above section.

### 1. Speech-to-Text Module:

This method takes the input language and output language as input from the user interface and once the method is called it takes speech input from the user in the input language given and converts to text by performing speech recognition on given audio. The given text is sent to translate method and gets the text in selected output language.

```
def speech_to_text(input_language,output_language):
    listener = sr.Recognizer()
    engine = pyttsx3.init()
    voices = engine.getProperty('voices')
    engine.setProperty('voice', voices[1].id)
    engine.setProperty("rate", 140)
    engine.say("Hi there! I am your converter. Please input your speech")
    engine.runAndWait()
    with sr.Microphone() as source:
        listener.adjust_for_ambient_noise(source)
        print("Listening...")
        audio = listener.listen(source)
        text = listener.recognize_google(audio, language=input_language)
        translation = translate(text, output_language)
    return translation
if st.button("🗣️"):
    output_text = speech_to_text(input_language,output_language)
    st.markdown(F'#Output Text:')
    st.write(f'{output_text}')
```

Figure 1: Illustrates the function for the Speech-to-Text

## 2. Text-to-Speech Module:

This method accepts the text and output language as input from the user interface and upon the execution of method it calls the translate method to convert the text to selected output language format and once it gets in the required translated text it converts the text to the selected audio output format.

```
def text_to_speech(input_text,output_language):
    translation = translate(input_text, output_language)
    synthesis_input = texttospeech_v1.SynthesisInput(text=translation)
    voice = texttospeech_v1.VoiceSelectionParams(
        language_code="en-In", ssml_gender=texttospeech.SsmlVoiceGender.FEMALE)
    # Select the type of audio file you want returned
    audio_config = texttospeech_v1.AudioConfig(
        # https://cloud.google.com/text-to-speech/docs/reference/rpc/google.cloud.texttospeech.v1#au
        audio_encoding=texttospeech_v1.AudioEncoding.MP3)
    # Perform text-to-speech request on the text input with the selected voice parameters and audio
    response = client.synthesize_speech(
        input=synthesis_input, voice=voice, audio_config=audio_config
    )
    # returning response's audio_content as binary.
    return response.audio_content,translation
display_output_text = st.checkbox("Display output text")
if st.button("convert"):
    audio_bytes,output_text=text_to_speech(text,output_language)
    st.markdown(f"## Your audio:")
    st.audio(audio_bytes, format="audio/mp3", start_time=0)
    if display_output_text:
        st.markdown(f"## Output text:")
        st.write(f"{output_text}")
```

Figure 2: Illustrates the function for the Text-to-Speech

### 3. Translate Module:

The translate method is a mediator between the text to speech or speech to text method where it takes text and output language as parameters and translates the text to the given output language and returns the translated text to the respective methods.

```
def translate(text, language):  
    from google.cloud import translate_v2 as translate  
    translate_client = translate.Client()  
  
    if isinstance(text, bytes):  
        text = text.decode('utf-8')  
  
    result = translate_client.translate(  
        text, target_language=language)  
  
    print(u'Text: {}'.format(result['input']))  
    print(u'Translation: {}'.format(result['translatedText']))  
  
    print(u'Detected source language: {}'.format(  
        result['detectedSourceLanguage']))  
    )  
    return result['translatedText']
```

Figure 3: Illustrates the function for Translate



#### 4. GUI Module:

In this module, a front-end environment is created by using the buttons, pup down options and input fields. In the below images we can see the snippets of the code for the UI features. And also, the representation of the result for each module respectively.

```
if action == 'text to speech':
    st.title("Text to speech")

    text = st.text_input("Enter text")

    out_lang = st.selectbox(
        "Select your output language",
        ("English", "Hindi", "Telugu", "Tamil", "Malayalam", "Bengali", "korean", "Chinese", "Japanese"),
    )
```

Figure 4: Illustrates the options from the UI

```
if st.button("convert"):
    audio_bytes,output_text=text_to_speech(text,output_language)
    st.markdown(f"## Your audio:")
    st.audio(audio_bytes, format="audio/mp3", start_time=0)
    if display_output_text:
        st.markdown(f"## Output text:")
        st.write(f" {output_text}")

if action == 'speech to text' :
    st.title("Speech to Text")
    in_lang = st.selectbox(
        "Select your input language",
        ("English", "Hindi", "Telugu", "Tamil", "Malayalam", "Bengali", "korean", "Chinese", "Japanese"),
    )
```

Figure 5: Illustrates the options from the UI

What would you like to do?

☒ speech to text

☐ text to speech

## Speech to Text

Select your input language

English

Select your output language

English




Figure 6: Illustrates the Streamlit UI for speech-to-text

What would you like to do?

☐ speech to text

☒ text to speech

## Text to speech

Enter text

Select your output language

English

☐ Display output text

convert

Figure 7: Illustrates the Streamlit UI for text-to-speech

## Results:

```

1  import speech_recognition as sr
2  import pyttsx3
3  import os
4  from google.cloud import texttospeech
5  from google.cloud import texttospeech_v1
6  import streamlit as st
7

```

### Packages Imported

```

Directory: C:\Users\gruha\Jupyter Notebook Workspace\Python Speech Recognition Streamlit

Mode                LastWriteTime         Length Name
----                -
-a----          2022-04-16   5:40 PM           6668 finalproject_testing.py
-a----          2022-04-16   5:40 PM           2330 keys.json
-a----          2022-04-16   5:40 PM           262 required_libraries.txt
-a----          2022-04-15   8:29 PM          9993 Voice List.csv

PS C:\Users\gruha\Jupyter Notebook Workspace\Python Speech Recognition Streamlit> python .\finalproject_testing.py
2022-04-17 23:24:49.940
Warning: to view this Streamlit app on a browser, run it with the following
command:

    streamlit run .\finalproject_testing.py [ARGUMENTS]
PS C:\Users\gruha\Jupyter Notebook Workspace\Python Speech Recognition Streamlit> streamlit run .\finalproject_testing.py

You can now view your Streamlit app in your browser.

Local URL: http://localhost:8501
Network URL: http://10.0.0.197:8501

```

### Steps Involved for Executing

Home page of our application displaying radio buttons to select either speech to text or text to speech.

What would you like to do?

- ☒ speech to text  
☐ text to speech

## Speech to Text

Select your input language

English

Select your output language

English



### Speech to Text Screen

Selection of speech to text button will display the speech to text module where it will ask for the input language and output language and upon clicking on mic button it will take speech as input and display the text output in selected output language.

What would you like to do?

- ☒ speech to text  
☐ text to speech

## Speech to Text

Select your input language

English

Select your output language

Hindi



#Output Text:

मैं स्पीच टू टेक्स्ट रिकग्निशन मॉड्यूल पर काम कर रहा हूँ

### Result on Speech to Text (English to Hindi)

Selecting the text to speech radio button will result in display of text to speech module.

What would you like to do?

☐ speech to text

☒ text to speech

### Text to speech

Enter text

Select your output language

Hindi

☐ Display output text

convert

### Text to Speech Screen

In the UI we are taking text input from the user and asking to select the output language in which the audio will be provided and also giving an option to display the text in selected language. The convert button execution provides the audio output for the given text.

This is python speech to text module

Select your output language

Telugu

☒ Display output text

convert

### Your audio:

▶ 0:00 / 0:02

🔊 ⋮

### Output text:

ఇది పైథాన్ స్పీచ్ టు టెక్స్ట్ మాడ్యూల్

Result on Text to Speech (English to Telugu)

## **Conclusions and Future Work:**

In this project we have implemented the speech to text and text to speech web application using speech recognition and google cloud API's. We would also like to improve the performance of the application by using API's which could lead to more accuracy of results and develop better user interface and develop into a complete application and host on one of the servers and make available to the end users.

## References:

1. <https://cloud.google.com/text-to-speech/docs/reference/rpc/google.cloud.texttospeech.v1#audioencoding>
2. <https://pypi.org/project/SpeechRecognition/>
3. <https://pypi.org/project/pyttsx3/>
4. <https://pypi.org/project/PyAudio/>
5. <https://pypi.org/project/streamlit/>
6. <https://stackoverflow.com/questions/52283840/i-cant-install-pyaudio-on-windows-how-to-solve-error-microsoft-visual-c-14>
7. <https://towardsdatascience.com/streamlit-hands-on-from-zero-to-your-first-awesome-web-app-2c28f9f4e214>
8. <https://stackoverflow.com/questions/55936179/trying-to-install-pyaudio-using-pip>
9. <https://medium.com/mlearning-ai/a-rough-guide-to-the-speech-to-text-landscape-cdde2c3eaf7e>
10. <https://cloud.google.com/speech-to-text/docs/basics>