

VISIVOICE

Voice for the dumb and spirit for the deaf

Mentor: Alok Yadav

Project duration: 26 Jan 2020 to 12 Feb 2020

Group Members:

Yash

Aniket Roy

Yash Jaitly

Aman Surkar

Omkar Solase

Omkar Rajale

Palak Gadekar

Ankit Gundewar

Shreyas Kulkarni

Guntreddi Harshavardhan

Name: Yash

Email: yash754311@gmail.com

Mob. No: +91 9125041358

ACKNOWLEDGEMENT

We the members of ML2/12B2 would like to thank Verzeo for having given us the opportunity and the guidance for this project. By making resources easily accessible to us, they made learning Machine Learning a breeze for us which helped us tremendously throughout this project.

INTRODUCTION

Ever since the beginning of life on earth, humankind has always been struggling to create his haven in nature. Be it fighting beast twice his size or foraging for food, shelter and clothing, man has always found it difficult to live by. But whatever may have been the circumstances, the one thing that remained with him throughout the millennia are his 5 senses which helped him communicate, coordinate and thrive in an environment that his body might not have been physically made for. Back in those times, even if a person lost one of his senses, it spelled certain doom for that person.

Today with the advent of the technological era and the onset of modern times, we have found technical solutions to almost every problem that mankind may have to face. This project is a Modern solution to bring voice to people who have lost their ability to hear it. Using modern technologies like Machine Learning and Natural Language Processing, we the members of batch ML2/12B2 bring to you our version of a “Speech to Text and Text to Speech Engine.”

Software used

- Jupyter notebook
- Google colab

SCOPE

The project is aimed at people who want to have a voice even if they naturally don't have one and at those who want to converse with others even when they do not possess the ability to hear.

WORKING

This **Speech to Text** engine utilises libraries like SpeechRecognition and pyAudio to convert either .wav files or real time audio to text format.

1. The SpeechRecognition interface of the Web Speech API is the controller interface for the recognition service; this also handles the Speech Recognition Event sent from the recognition service. Library for performing speech recognition, with support for several engines and APIs, online and offline.
 - Speech recognition engine/API support:
 - CMU Sphinx (works offline)
 - Google Speech Recognition
 - Google Cloud Speech API
 - Wit.ai
 - Microsoft Bing Voice Recognition
 - Houndify API
 - IBM Speech to Text
 - Snowboy Hotword Detection (works offline)

We have used the Google Speech Recognition API for the project.

2. PyAudio provides Python bindings for PortAudio, the cross-platform audio I/O library. With PyAudio, you can easily use Python to play and record audio on a variety of platforms, such as GNU/Linux, Microsoft Windows, and Apple Mac OS X / macOS./

We have used PyAudio Library from PyPi for preprocessing and processing of the audio into a favourable format (txt). We also use it to display the transcript of the library directly in the console.

3. Our **Text to Speech** engine utilises libraries like OS, gTTS (Google Text to Speech) and IPython to either save the text into a music file or instantaneously play the file created.

➤ **OS:**

This module provides a portable way of using operating system dependent functionality.

➤ **gTTS:**

gTTS (Google Text-to-Speech), a Python library and CLI tool to interface with Google Translate' s text-to-speech API. Write spoken mp3 data to a file, a file-like object (bytestring) for further audio manipulation, or stdout.

gTTS is one of the libraries included in the PyPi repository and consists of very easy to use tools for converting text to audio. The gTTS API supports several languages including English, Hindi, Tamil, French, German and many more. The speech can be delivered in any one of the two available audio speeds, fast or slow.

CODE EXPLANATION

The code has three functions to accomplish given four tasks. These functions are-

1. text_to_speech():

text to speech conversion model

```
In [2]: def text_to_speech():
        print("Enter text")
        x = input()
        tts = gTTS(text=x, lang='en')      # google text to speech api
        tts.save("sample.wav")           # save the output in the file sample.wav
```

Output

```
1. Speech to text
2. Text to speech
3. Exit
enter your choice: 1
Speak Anything :
You said : hello this is sample audio
```

2. speech_to_text():

speech to text conversion model

```
In [3]: def speech_to_text():
        r = sr.Recognizer()              # initialize recognizer
        with sr.Microphone() as source:  # mention source it will be either Microphone or audio files.
            print("Speak Anything :")
            audio = r.listen(source)      # listen to the source
            try:
                text = r.recognize_google(audio) # use recognizer to convert our audio into text part.
                print("You said : {}".format(text))
            except:
                print("Sorry could not recognize your voice")
```

```
1. Speech to text
2. Text to speech
3. Exit
enter your choice: 2
Enter text
hey this is the major project we are working upon
```

play output sound file

```
In [7]: Audio(sound_file, autoplay=True)
```

Out[7]:



3. menu ():

choices

```
In [4]: def menu (choice):
        if choice is '1':
            speech_to_text()
        elif choice is '2':
            text_to_speech()
        elif choice is 3:
            exit()
        else:
            print("Wrong choice!\n Please choose from options.")
```

SPEECH TO TEXT OUTPUTS

1. Playing a song to the engine:

Recognizing Speech in the Audio

Finally, you can call `recognize_google()` to perform the transcription.

```
In [9]: a=r.recognize_google(audio)
```

```
In [10]: print(a)
```

worlds no sleeping without screen pain online now wonders think your so cream in now you say something by means we say thank you please at you I want pena morning too much so so so do you like it you are typing that I am bad guy if you tell if you know that don't need to play your mama I like to sing along with me but she want sing this song is willing to you with love

```
In [ ]:
```

2. Reading out a passage to the engine:

Recognizing Speech in the Audio

Finally, you can call `recognize_google()` to perform the transcription.

```
In [9]: a=r.recognize_google(audio)
```

```
In [11]: print(a)
```

voice recorder is it convenient and simple online tool that can be used to write in your browser allows you to record your voice using a microphone and save it as a MP3 file free to use voice recorder is completely free no hidden payment activation fees for charges for extra auto silent streaming voice record automatically detect silent fragments at the beginning and the end of the recording and delete them for your convenience cut your recording after the recording is completely can crop it if the selection you actually need it

TEXT TO SPEECH OUTPUTS

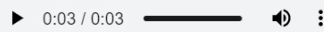
1. Instantly playing typed text:

```
In [1]: from gtts import gTTS
import os
import IPython.display as ipd
from IPython.display import Audio
sound_file = "C:\GTTS\Good.wav"
```

```
In [2]: print("Enter text")
x = input()
tts = gTTS(text="      "+x,lang='en')
tts.save("C:\GTTS\Good.wav")
Audio(sound_file, autoplay=True)
```

Enter text
Ok Ankit, What can I do for you today?

Out[2]:



2. Saving the typed text in any audio format:

```
In [1]: from gtts import gTTS
import os
import IPython.display as ipd
from IPython.display import Audio
sound_file = "C:\GTTS\Good.wav"
```

```
In [3]: print("Enter text")
x = input()
tts = gTTS(text="      "+x,lang='en')
tts.save("C:\GTTS\Good.mp3")
#Audio(sound_file, autoplay=True)
```

Enter text
Hey there!

PROGRAM/ SOURCE CODE: https://github.com/yash0429/ML_project

REFERENCES

1. <https://medium.com/@rahulvaish/speech-to-text-python-77b510f06de>
2. <https://www.simplifiedpython.net/speech-recognition-python/>