

AI PROJECT WITH PYTHON

ON

SPEECH RECOGNITION

INT404



L OVELY
P ROFESSIONAL
U NIVERSITY

Transforming Education Transforming India

Submitted By-
DIVYANSHU – 11802034
ANIKET – 11805162
AMAN - 11804394
VISHAL - 11804269
SAURAV – 11802301

Submitted to-
ANKITA WADHAWAN

INTRODUCTION

AI with Python – Speech Recognition



Speech is the most basic means of adult human communication. The basic goal of speech processing is to provide an interaction between a human and a machine.

Speech processing system has mainly three tasks –

- **First**, speech recognition that allows the machine to catch the words, phrases and sentences we speak
- **Second**, natural language processing to allow the machine to understand what we speak, and
- **Third**, speech synthesis to allow the machine to speak.

Building a Speech Recognizer-

Speech Recognition or Automatic Speech Recognition (ASR) is the center of attention for AI projects like robotics.

Difficulties in developing a speech recognition system

- **Size of the vocabulary**

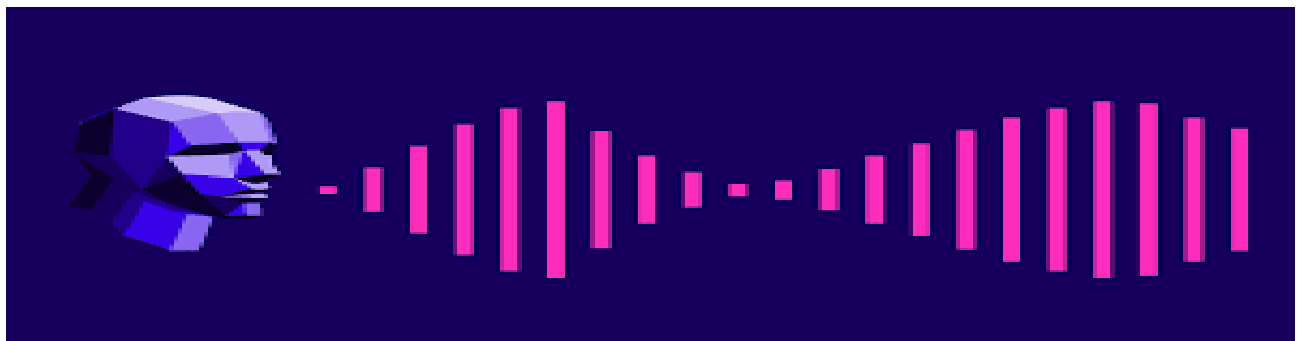
- Channel characteristics
- Speaking mode
- Speaking style
- Speaker dependency
- Type of noise
- Microphone characteristics

Visualizing Audio Signals - Reading from a File and Working on it

This is the first step in building speech recognition system as it gives an understanding of how an audio signal is structured. Some common steps that can be followed to work with audio signals are as follows –

Recording

Sampling



Recognition of Spoken Words -

Speech recognition means that when humans are speaking, a machine understands it. Here we are using Google Speech API in Python to make it happen. We need to install the following packages for this –

- **Pyaudio** - `pip install Pyaudio` command.

- **SpeechRecognition** - pip install **SpeechRecognition**.
- **Google-Speech-API** - pip install **google-api-python-client**.

Characterizing the Audio Signal: Transforming to Frequency Domain-

Characterizing an audio signal involves converting the time domain signal into frequency domain, and understanding its frequency components, by. This is an important step because it gives a lot of information about the signal. You can use a mathematical tool like Fourier Transform to perform this transformation.

Generating Monotone Audio Signal-

The two steps that you have seen till now are important to learn about signals. Now, this step will be useful if you want to generate the audio signal with some predefined parameters. Note that this step will save the audio signal in an output file.

Feature Extraction from Speech-

This is the most important step in building a speech recognizer because after converting the speech signal into the frequency domain, we must convert it into the usable form of feature vector. We can use different feature extraction techniques like MFCC, PLP, PLP-RASTA etc. for this purpose.

Code-Snippet

```
import pyttsx3 #pip install pyttsx3
import speech_recognition as sr #pip install speechRecognition
import datetime
import wikipedia #pip install wikipedia
import webbrowser
import os
#import smtplib

|
engine = pyttsx3.init('sapi5')
voices = engine.getProperty('voices')
# print(voices[1].id)
engine.setProperty('voice', voices[0].id)

def speak(audio):
    engine.say(audio)
    engine.runAndWait()

def wishMe():
    hour = int(datetime.datetime.now().hour)
    if hour>=0 and hour<12:
        speak("Good Morning!")
```

```
elif hour>=12 and hour<18:
    speak("Good Afternoon!")

else:
    speak("Good Evening!")

speak("I am selfmade Sir. Please tell me how may I help you")

def takeCommand():
    #It takes microphone input from the user and returns string output

    r = sr.Recognizer()
    with sr.Microphone() as source:
        print("Listening...")
        r.pause_threshold = 1
        audio = r.listen(source)

    try:
        print("Recognizing...")
        query = r.recognize_google(audio, language='en-in')
        print(f"User said: {query}\n")

    except Exception as e:
```

```

except Exception as e:
    = print(e)
    print("Say that again please...")
    return "None"
return query

#def sendEmail(to, content):
#server = smtplib.SMTP('smtp.gmail.com', 587)
#server.ehlo()
#server.starttls()
#server.login('youremail@gmail.com', 'your-password')
#server.sendmail('youremail@gmail.com', to, content)
#server.close()

if __name__ == "__main__":
    wishMe()
    while True:
        # if 1:
        query = takeCommand().lower()

        # Logic for executing tasks based on query
        if 'wikipedia' in query:
            speak('Searching Wikipedia...')
            query = query.replace("wikipedia", "")

```

```

            speak(' Searching Wikipedia... ')
            query = query.replace("wikipedia", "")
            results = wikipedia.summary(query, sentences=2)
            speak("According to Wikipedia")
            print(results)
            speak(results)

        elif 'open youtube' in query:
            webbrowser.open("youtube.com")

        elif 'open google' in query:
            webbrowser.open("google.com")

        elif 'open stackoverflow' in query:
            webbrowser.open("stackoverflow.com")

        elif 'play music' in query:
            music_dir = 'D:\\Non Critical\\songs\\Favorite Songs2'
            songs = os.listdir(music_dir)
            print(songs)
            os.startfile(os.path.join(music_dir, songs[0]))

        elif 'the time' in query:

```

```

music_dir = Path('D:\Non critical songs\Favorite Songs2')
songs = os.listdir(music_dir)
print(songs)
os.startfile(os.path.join(music_dir, songs[0]))

elif 'the time' in query:
    strTime = datetime.datetime.now().strftime("%H:%M:%S")
    speak(f"Sir, the time is {strTime}")

elif 'open code' in query:
    codePath = Path("C:\\Users\\Divyanshu\\AppData\\Local\\Programs\\Micros
os.startfile(codePath)

elif 'stop' in query or 'bye' in query:
    speak("Quitting sir thanks for your time")
    exit()

elif 'train' in query:
    speak("Please tell the train number")
    search = takeCommand()
    df = pd.read_excel(Path('C:/Users/Divyanshu/Documents/Train.xlsx'))
    print(df)

```

```

strTime = datetime.datetime.now().strftime("%H:%M:%S")
speak(f"Sir, the time is {strTime}")

elif 'open code' in query:
    codePath = "C:\\Users\\Haris\\AppData\\Local\\Programs\\Microsoft V
os.startfile(codePath)

I
#elif 'email to Vishal' in query:
    #try:
        #speak("What should I say?")
        #content = takeCommand()
        #to = "yourEmail@gmail.com"
        #sendEmail(to, content)
        #speak("Email has been sent!")
    #except Exception as e:
        #print(e)
        #speak("Sorry ")

```


Methodology

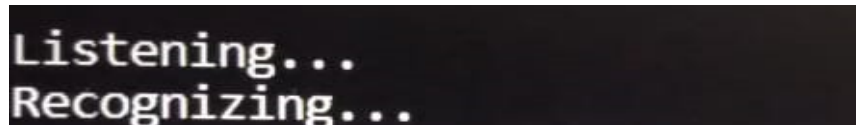
Working of the Project.

Speak- The assistant will speak the following introduction, the output and the following things according to which good is given. It will use the laptop microphone to hear the input from the user and later recognizing the voice said by user and match the words is code and if anything match it will show the output.

Wish Me-The assistant will speak the Message included in the introduction even it will wish the morning afternoon and even the evening depending upon the real time based scenario. It will wish the morning from 04HH to 11HH 59MM. It will wish the afternoon from 12HH to 17HH 59MM. It will wish the evening from 18HH to 03HH 59MM.

Take Command- The assistant will take microphone(speech) input from the user and returns string output. It will be sub-divide into many different part as described below.

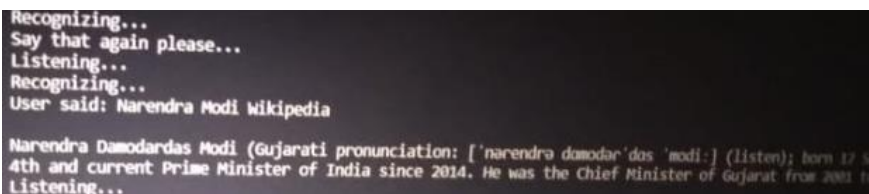
Listening-The assistant will open the microphone and try to hear what the user wants to convey to it.



```
Listening...
Recognizing...
```

Recognizing- The assistant will try to recognize the input spoken by the user and then check the code whether the word that it recognized by the assistant is there or not if the input matches it will show the output otherwise it will speak “Say that again please” this line which means to give the input again by the user. If the word is correctly recognized, it will follow the instructions assigned to it.

Wikipedia- If the word is recognized as “Wikipedia” it will search the Wikipedia according to the input given by the user. E.g. if we say Narendra Modi Wikipedia so assistant will speak “searching Wikipedia Narendra Modi” and then after it “According to Wikipedia...” and the details of that particular person.



```
Recognizing...
Say that again please...
Listening...
Recognizing...
User said: Narendra Modi Wikipedia

Narendra Damodardas Modi (Gujarati pronunciation: [ˈnəɾɐndɾə dɑmɔdɑɾˈdɑs ˈmoːɖi] (listen); born 17 September 1950) is an Indian politician, leader of the Bharatiya Janata Party (BJP) and current Prime Minister of India since 2014. He was the Chief Minister of Gujarat from 2001 to 2003.
Listening...
```


You tube- If the word is recognized as “YouTube”, it will open the internet explorer and directly start opening the default web browser by the link “youtube.com”.

```
Listening...
Recognizing...
User said: open YouTube
```

Google- If the word is recognized as “Google”, it will open the internet explorer and directly start opening the Google by the link “google.com”.

```
Listening...
Recognizing...
User said: open Google
```

Train Information- If the word is recognized as “Train info”. It will fetch the detail from csv file and returns the detail of all the train and display it on the terminal.

```
Listening...
Recognizing...
User said: train info
```

	Train No	Train Name	Source	Destination	Departure Time
0	12557	Saptkranti	Muzaffarpur	Delhi	12:00:00
1	12558	Satyagrah	Rexaul	Delhi	10:00:00

Stack Overflow- If the word is recognized as “Stack Over Flow” it will open the internet explorer and directly start opening the Stack Over Flow website by the link “stackoverflow.com”.

```
Say that again please...
Listening...
Recognizing...
User said: open stackoverflow
```

Play Music- If the word is recognized as “Play Music” it will search the .mp3 or .mp4 file in the default path of the device the is provided by the programmer in the programming. E.g. if we say if we say Play Music so assistant will search in the path like “D:\\Non Critical\\songs\\Favourite Songs2” and it will play that particular song.

```
Listening...
Recognizing...
Say that again please...
Listening...
Recognizing...
User said: play music
```

[*02. Samne Yeh Kaun Aaya (The Unwind Mix) [Ash King] 190kbps.mp3]

The Time- If the word is recognized as “The Time” it will check the real time from the device and speak the same in terms of “HH:MM:SS”. E.g. if we say the time so assistant will check the time and if the time is 08:14:21 P.M. it will speak “Sir, the time is 20HH:14MM:21SS”.

```
PS C:\Users\Divyanshu> & C:/Users/Divya
Listening...
Recognizing...
User said: the time
Sir, the time is 12:15:55
Listening...
Recognizing...
User said: open YouTube
Listening...
```

Open Code- If the word is recognized as “Open Code” it will search the .java or .py file in the default path of the device the is provided by the programmer in the programming. E.g. if we say if we say Open Code so assistant will search in the path like “C:\\Users\\XYZ\\AppData\\Local\\Programs\\project.py” and it will open the code.

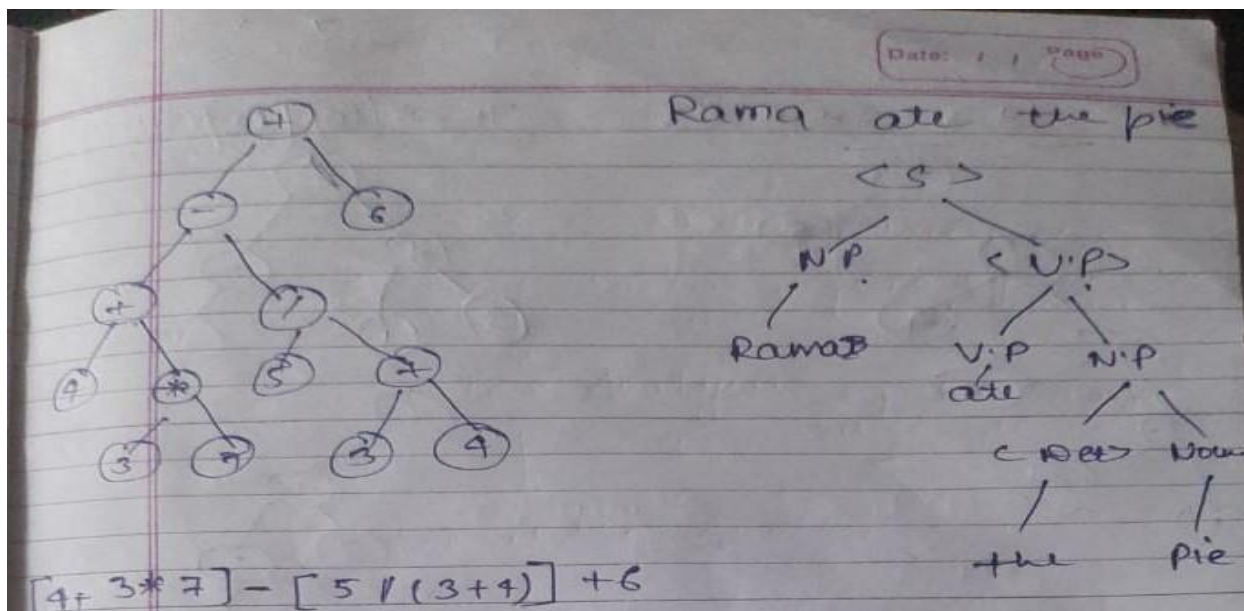
Stop- If the word is recognized as “Stop” it will speak “Quitting sir thanks for your time” and the codes terminates.

```
Listening...
Recognizing...
User said: bye
Quitting sir thanks for your time Have a good day
Divyanshu
```

Algorithms used in Speech Recognition

- NLP (Natural Language Processing) & Tokenization
- PLP
- Deep Neural Networks
- Discrimination training
- WFST Frameworks etc;

Natural Language Processing (NLP) – It is a subfield of computer science, artificial intelligence, information engineering, and human-computer interaction. This field focuses on how to program computers to process and analyze large amounts of natural language data. It is difficult to perform as the process of reading and understanding languages is far more complex than it seems at first glance.



Tokenization – It is the process of tokenizing or splitting a string, text into a list of tokens. One can think of token as parts like a word is a token in a sentence, and a sentence is a token in a paragraph.

Sentence Tokenization – Splitting sentences in the paragraph

The **sent_tokenize** function uses an instance of **PunktSentence Tokenizer** from the **nltk.tokenize.punkt** module, which is already been trained and thus very well knows to mark the end and beginning of sentence at what characters and punctuation.

Example:

```
from nltk.tokenize import sent_tokenize
text = "Hello everyone. You are studying NLP article"
sent_tokenize(text)
```

Output: ['Hello everyone.', 'Welcome to GeeksforGeeks.',
'You are studying NLP article']

WordPunctTokenizer- It separates the punctuation from the words.

Example:

```
from nltk.tokenize import WordPunctTokenizer

tokenizer = WordPunctTokenizer()

tokenizer.tokenize("Let's see how it's working.")
```

Output: ['Let', "'", 's', 'see', 'how', 'it', "'", 's', 'working', '.']

- **Perceptual linear prediction (PLP)-** It technique combines the critical bands, intensity-to-loudness compression and equal loudness pre-emphasis in the extraction of relevant information from speech.

It is rooted in the nonlinear bark scale and was initially intended for use in speech recognition tasks by eliminating the speaker dependent features. It uses linear predictions for spectral smoothing, hence, the name is perceptual linear prediction. PLP is a combination of both spectral analysis and **linear prediction** analysis.

- With the emergence of **Deep Learning neural networks** were are used in many aspects of speech recognition such as phoneme classification, isolated word recognition, audiovisual speech recognition, audio-visual speaker recognition and speaker adaptation.

Deep Learning enables the development of Automatic Speech Recognition (ASR) systems. These ASR systems require separate models, namely acoustic model (AM), a

pronunciation model (PM) and a language model (LM). The AM is typically trained to recognize context-dependent states or phonemes, by bootstrapping from an existing model which is used for alignment. The PM maps the sequences of phonemes produced by the AM into word sequences. Word sequences are scored using LM trained on large amounts of text data, which estimate probabilities of word sequences.

- **Discrimination learning** – It is defined in psychology as the ability to respond differently to different stimuli. This type of learning is used in studies regarding operant and classical conditioning. The complex and fundamental nature of discrimination learning allows for psychologists and researchers to perform more in-depth research that supports psychological advancements. Research on the basic principles underlying this learning style has their roots in neuropsychology sub-processes.
- **Weighted Finite-State Transducers (WFST)**- It is good at modeling HMM and solving state machine problems. In the ASR context, it provides a natural representation for HMM phonetic models, context-dependent phones, pronunciation lexicons, and grammars.

Different Python Modules used in this Project.

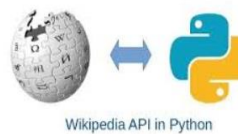


1. **Date Time Module- Saurav**

The date time module supplies classes for manipulating dates and times. Common features include:

- 1.Objects of these types are immutable.
- 2.Objects of these types are hashable, meaning that they can be used as dictionary keys.
- 3.Objects of these types support efficient pickling via the pickle module.

I have used this module in order to record the date and time when the user speaks something.



2. **Wikipedia Module- Aniket**

Wikipedia is a Python library that makes it easy to access and parse data from Wikipedia. You can Search Wikipedia, get article summaries, get data like links and images from a page, and much more.

I have used this module in order to retrieve data from Wikipedia with the help of web browser module.



3. **Web Browser Module- Aman**

The web browser module provides a high-level interface to allow displaying Web-based documents to users. Under most circumstances, simply calling the `open()` function from this module will open url using the default browser . You have to import the module and use `open()` function. This module is provided in order to fetch the information from web browser The web browser module provides a high-level interface to allow displaying Web-based documents to users. Under most circumstances, simply calling the `open()` function from this module will open url using the default browser . You have to import the module and use `open()` function. This module is provided in order to fetch the information from web browser.



4. **OS Module- Vishal**

The OS module in Python provides a way of using operating system dependent functionality. The functions that the OS module provides allows you to interface with the underlying operating system that Python is running on – be that Windows, Mac or Linux. You can find important information about your location or about the process

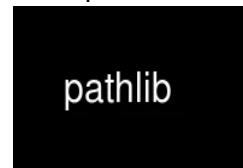
This module is used in order to make our program perform system functionality.



5. Pyttsx3 and Pyaudio Module- Divyanshu

pyttsx3 is a text-to-speech conversion library in Python. Unlike alternative libraries, it works offline and is compatible with both Python 2 and 3. An application invokes the `pyttsx3.init()` factory function to get a reference to a `pyttsx3`. Engine instance. it is a very easy to use tool which converts the entered text into speech. The `pyttsx3` module supports two voices first is female and the second is male which is provided by “sapi5” for windows.

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. This Module are used in order to provide speech support to the program and for speech input.



6. Pathlib Module- Saurav

This is simple module used in order to remove the path error from the program.



7. Pandas Module- Aman

Pandas is an open source library in Python. It provides ready to use high-performance data structures and data analysis tools. Pandas module runs on top of NumPy and it is popularly used for data science and data analytics.

This module is used in order to fetch the train detail from csv file and display the searched train information as according to the user.



8. Speech Recognition Module- All Member

Speech Recognition is an important feature in several applications used such as home automation, artificial intelligence, etc. This article aims to provide an introduction on how to make use of the Speech Recognition library of Python. This is useful as it can be used on microcontrollers such as Raspberri Pi with the help of an external microphone.

The following must be installed-:

1. `sudo pip install SpeechRecognition`.
2. `sudo apt-get install python-pyaudio python3-pyaudio` or `pip install pyaudio`.

This is most important module in your project as it provides the main functionality in our project to convert speech into text.

Drawbacks of Speech Recognition

Limited Vocabulary-

While voice recognition technology recognizes most words in the English language, it still struggles to recognize names and slang words. With the limited vocabulary of speech recognition devices, it might not be worth the purchase if you're continually having to go back over your work and fix many mistakes.

Delays-

Voice recognition devices have been designed to help you speed up your work, but they are prone to mistakes and mishaps. These devices often take a bit to register what's being said, which can be frustrating and interrupt your thought flow. Having many frequent pauses can easily put you in a bad mood and when it glitches, it can force you to have to abandon the technology to get your work done.

Background Noise Interference-

To get the best out of voice recognition software, you need a quiet environment. Systems don't work so well if there is a lot of background noise. They may not be able to differentiate between your speech, other people talking and other ambient noise, leading to transcription mix-ups and errors. This can cause problems if you work in a busy office or noisy environment. Wearing close-talking microphones or noise-canceling headsets can help the system focus on your speech.

Future Scope

Speech recognition is a thriving domain with many important applications. It's easy to predict that speech recognition research will continue as well as important practical applications will be created. Accurate speech recognition is not so hard problem so it should be solved in a foreseeable future. And it's not about AI because it's obvious that most of the speech recognition issues are not caused by the lack of understanding but rather a lack of good algorithms.

Noises, accents and so on are just purely technical problems which will be eventually solved. Researches often consider speech recognition in a noisy environment as a standalone problem with a practical goal to build an application that works. At the same time our knowledge about speech fundamentally improves from day to day and the goals are more and more ambitious. Recent BABEL programs aims to improve support for non-English languages and it's planned that we will have to quite good step forward in a next few years. Some leading researchers are working on language-independent speech recognition. The accuracy on the standard test sets also improves from year to year. And voice applications are already in every smartphone.

This specific area of AI, end up being productive in each specialized field. We have additionally actualized this so as to show how it is valuable in various field as we have made a little undertaking to exhibit it use in various documented, for example, railroad, looking through feed and so on;

Like PCs began to play chess better than human, speech recognition before long will be improved by PCs as well. Critically, that will include some significant information about nature in general and human mind specifically. So speech recognition is a significant advance to our investigation of the nature laws. Our venture can be utilized by railroads and other center point to show distinctive data utilizing speech recognition.

References

1. <https://www.quora.com/>
2. <https://www.python.org/>
3. <https://www.geeksforgeeks.org/wikipedia-module-in-python/>
4. <https://www.wikipedia.org/>
5. <https://www.youtube.com/>
6. <https://www.w3schools.com/python/>
7. https://www.tutorialspoint.com/python/python_pdf_version.htm