Project Report

On

SPEECH RECOGNITION USING PYTHON

Project report submitted In partial fulfillment of the requirement for the degree of Bachelor of Technology

In

Computer Science and Engineering Information Technology

By

AKSHIT Chaudhary [171271]

Rahul Tomar [171295]

To

Pradeep Kumar Khokhar

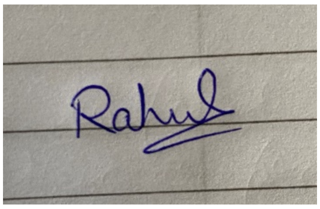
Department of Computer Science & Engineering and Information Technology

Jaypee University of Information Technology Waknaghat, Solan-173234,

Himachal Pradesh

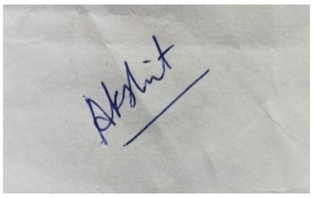
CERTIFICATE

This is to certify that the work in this Project title as "**AN IMPLEMENTATION OF SPEECH RECOGNITION**" is entirely written, successfully completed and demonstrated by the following students themselves as a fulfillment of requirement for Bachelor's of Engineering in Computer Science

Sign: 

Student Name: Rahul Tomar

Roll no: 171295

Sign: 

Student Name: Akshit Chaudhary

Roll no: 171271

Supervisor Name: Dr. Pradeep Kumar

Designation: Associate Professor (Senior Grade)

Department name: Computer Science and Engineering and Information technology

Dated:

ACKNOWLEDGEMENTS

I would like to express my deepest appreciation to all those who have been helping me throughout the project and without whom this project would have been a very difficult task. I would like to thank all of them.

I am highly indebted to Dr Pradeep Kumar Khokhar for his guidance and constant supervision as well as for providing necessary information regarding the project & also for their support in doing my project. I would like to express my gratitude towards members of JUIT for their kind co-operation and encouragement which helped me in doing this project. My thanks and appreciations also go to our colleagues who have helped me out with their abilities in developing the project.

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

TABLE OF CONTENTS

LIST OF GRAPH

LIST OF FIGURES

ABSTRACT

Speech recognition technology is one from the fast growing engineering technologies. It has a number of application in different areas and provides potential benefits. Nearly 20% people of the world are suffering from various disabilities; many of them are blind or unable to use their hands effectively.

The speech recognition systems in those particular cases provide a significant help to them, so that they can share information with people by operating computer through voice input.

This project is designed and developed keeping that factor into mind, and a little effort to achieve this aim. Our project is capable to recognize the speech and convert into text.

CHAPTER 1

1.1 INTRODUCTION

There is huge development in speech recognition technologies from last years as it had completely brought up huge progress based on the new machine learning algorithms.the speech recognition system proves to be benificial in many aspects as it reduces the wastage of time as well as helps the disabled individuals.

Speech technology with fields within the scope of the paper are to be presented in Fig. as the unified framework that encompasses covered topics, showing their complementarity, ranges and borders, interconnections, and intersections in the interdisciplinary area of Speech.

Unified framework

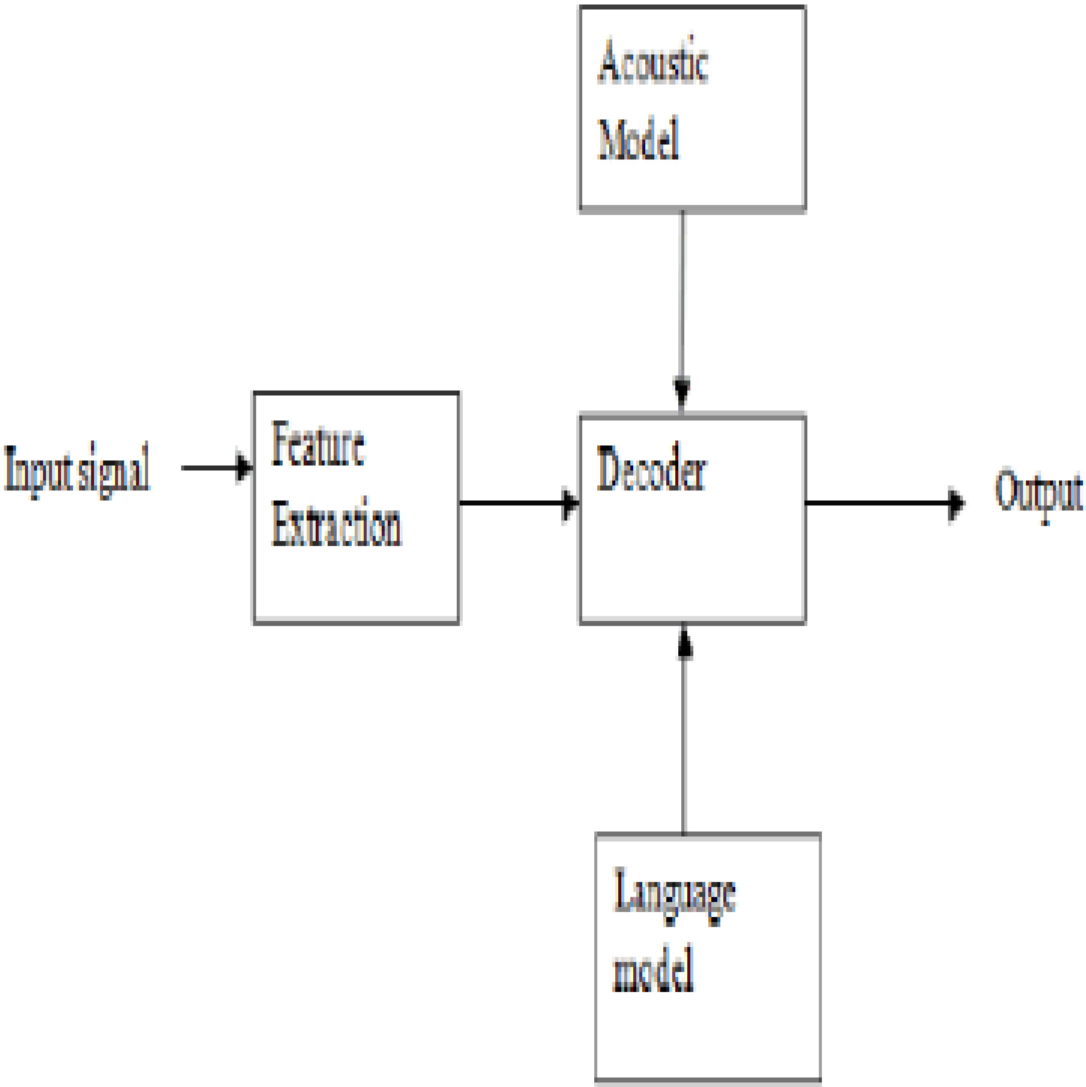


Fig: 1.1 Unified Framework

In mostly areas of the country, there are lot of people who don't know how to write and also how to read any word, so this project is very helpful for these type of people as you know in today's world

Everybody has its own mobile phones and they want to search a lot of things. In this project, they usually speak what they want to search and various results of such type opens in the browser window.

In this project, we made our machine recognise the speech passed as the audio file as well as the disection of the speech basis on the requirement .

Our Aim is to make the search fast and efficient and also reliable for every person by implementing basic search commands and also correct their vocabulary easily and also further implementing speaking mode like Siri in iphone’s.

1.2 **Objectives**

I. To be familer with the speech recognition and its fundamental.

II. Its working and application in different areas.

III. To implement it as an application for relative searches.

IV. Software which can be used for:

a) Speech recognition
b) Web searches
c) Word guessing

1.3 Problem Statement

Speech recognition is the process that recognizes all words being said by humans and to convert this speech into text and to analyse this texxt to produce the results required by the humans. The performance of this system majorly depends upon number of factors such as the speed of the spoken words by the user,vocabularies and the background noise caused by the environment .The speech recognition library of the package provided by the pypi library can be helpful in reducing various factors such as background noise which then makes the speech good for processing and the performing the tasks provided to this system such as words recognition ,web searches .

1.4 Methodology:

Due to the daily changes and enhancement in technology, not everyone is familiar with speech recognition technology.

The basic function of both speech synthesis and speech recognition is easy to understand as there are many powerful capabilities provided by speech recognition technology that helps many developers to understand and utilize this technology.

Despite the substantial growth and research in speech recognition technology their are still more limitations in this technology. Because of the speech recognition humans are able to utilize the time in various aspects and also it proves to be benificial to various disabled peoples,still this system is unfamilar with natural human to human conversations.

The complete knowledge of the limitation also the strength is very important for the accurate use of speech recognition technologies as there may be differences in the output provided by the system and the output required by the user for a particular input.Due to this understanding the user or developers of these application can make a decisions about whether the technology will benefit the use of speech-to-text in a particular speech input.

1.5 Scope

The speech recognition system in this project has the capability which could be same as the systems used by Iphones and google but cannot be as much effective as the functions provided by these systems.
This project is the basic implementation of speech to text conversion and also performing the basic tasks provided by the user to the system.

CHAPTER 2

(LITERATURE REVIEW)

2.1 HISTORY

The First speech recognition system were focused on numbers, not words. In 1952 bell Laboratory designed the “Audrey System” which could recognize a single voice speaking digits aloud. Ten years later IBM introduced “shoebox” which understood 16 words in English .

Across the globe other nations developed hardware that could recognize sound and sleep. And by the end of ‘60s , the technology could support words with 4 vowels and nine consonants.

1970’S

Speech recognition made several meaningful advancements in this Decade. This was mostly due to the US Department of defence and DARPA. The Speech Understanding Program SUR program ther ran was one of the largest of its kind in the history of speech recognition. Mellon ‘Harpy Speech System‘ came from this programand was capable of understanding over 1000 kind words that is about the same a three year old’s vocabulary.

Also significant in the 70’s was Bell Laboratories introduction od the system that could interpret Multiple voices.

1980’s

The ‘80s saw speech Recognition vocab go from few of hundreds words to the several thousands words. One of the Breakthroughs that came from a statistical methods known as The ‘ Hidden Markov Model0 ‘HMM’ ‘ . Instead of just using words and looking for the sound patterns. The Hmm estimated the probability of the unknown sounds actually being words .

1990’s

Speech recognition was propelled forward in the 90s in the large part because of the own personal computer. The faster processors made it possible for software like dragon dictate to become the more widely used bell south introduced the Voice Portal (VAL) in which was a dial in interactive voice recognition system . This System give new birth to the myriad of the phones tree system that are still in the existence Today.

2000s

From the year 20002 Speech recognition Technology had achieved close to the 80 percent accuracy.

For almost of all the Decade There aren't a lot of Advancements till google has come with a start of google search voice.

As it was an application which put speech recognition into hands of lakhs of people .

This was also Significant because that the processing power would be offloaded to its data Centres.

Not only for that, Google Application was collecting data from many billions of the searches which could help this to predict what a human is actually Saying.

That time Google's English voice search system, included 240 billion words from user searches.

2010s

In 2012 Apple Launched SIRI which was as same as the Google's VOICE SEARCH.

The early part of the decade saw an explosion of the other voice Recognition Applications.

And with Amazon's ALEXA, Google Home we've seen consumers Becoming More and More comfortable talking to Machines.

Today, Some of the Largest Technical Companies are competing to herald the speech accuracy title. In 2015, IBM achieved a word ERROR RATE pf 6.8%.

IN 2016 Microsoft overpassed IBM with a 5.8 % claim. Shortly After that IBM improved their Rate to 5.4 %. However it’s Google that claims the lowest Ratio rate at 4.8percent.

The Future

The tech to support speech Applications is today both Relatively Inexpensive and Powerful. With the betterment or the advance tech in Artificial Intelligence and to the increase amounts of Speech Data that can be easily mined, it is now possible to that voice becomes the next Dominant Interface.

At Sonix, We can also applause the many companies before us that propelled speech Recognition to where it is Today. We Automate Transcription workflow and make it fast , easy and more affordable.

We wouldn’t do this without the proper Work that has to been done before we.

Analysis:

From apple SIRI to Smart Devices of home, Speech Recognition Is very drastically used in our lives. This Speech Recognition project is to Utilize Kaggle Speech Recognition Challenge Dataset to Create Keras Model on above of tenserflow & to create predictions in the voice files.

Data Indigestion and Processing

Similar to image Recognition, the most important part of the speech Recognition is to convert audio into 2*2 Arrays.

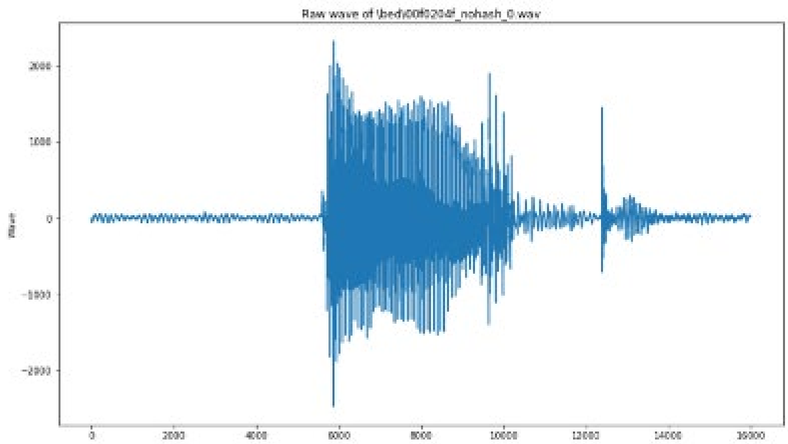
Sample Rate and raw Wave of audio Files:

Sample Rate of an Audio File represents the numbers of samples of Audio Carried per Second and is measured in Hz. The following image shows the relationship between the audio Raw Wave and Sample Rate of “Bed” audio file:

```
# visualize this audio wave:
fig = plt.figure(figsize=(14, 8))
plt.plot(samples)
plt.title('Raw wave of ' + filename)
plt.ylabel('Wave')
plt.show()
```

```
16000
[ -8 -11 -11 ...  13  10  16]
```

```
train_audio_path = 'C:\\Users\\...\\train\\audio'
filename = '\\bed\\00f0204f_nohash_0.wav'
sample_rate, samples = wavfile.read(str(train_audio_path) + filename)
ipd.Audio(samples, rate=sample_rate)
print(sample_rate)
print(samples)
```



2.3 SPEECH RECOGNITION TYPES

SPEECH RECOGNITION SYSTEM is basically Divided into following depending on various types:

Speaking Mode:

Basically it means that how the words are been spoken as in connected or in isolated. In Isolated word of speech Recognition System needs that speaker take pause between the words he speak. It means single kind word In connected word of speech recognition system did not need that the speaker take pause briefly in between the words. It generally means full length sentences in which words are then artificially keep away by silence.

Speaking Style:

Generally it Includes whether that the speech is in continuous form of spontaneous form. Continuous form is that spoken in natural form. Systems are to evaluated on speech read from the scripts that are prepared where as in spontaneous or extemporaneously generated, speech does not contain fluencies, and it is also difficult to figure out that speech read from the written script. It is also vastly much more hard as it tends to be peppered with unfluency like “uuh” and “uum”, no full sentaces, spluttering , stuttering, sneezing , cough, and also vocabulary is essentially ulimited, So there must be training to system to be able to tackle with unknown and hidden words.

Vocabulary :

IT is much simple to discriminate a smaller set of the words, but rate of error incareses as the size of the vocabulary increases.

For ex: 10 digits start from 0 -9 can easily be recognised rightly on the other side vocabulary whose size is 100 , 4000 or 15000 have the rate of error as 3%, 6%, 40% . The vocabulary is hard to predict or recognize if it contains Confused kind of words.

Enrollment:

This is kind of 2 ways

1)Speaker Dependent 2) speaker independent

In speaker dependent the user must be providing various samples of her or his speech before they’re used, a speaker dependent system is meant for use of only single kind speaker , where as speaker independent system is allowed or intended to use any type or kind of speaker

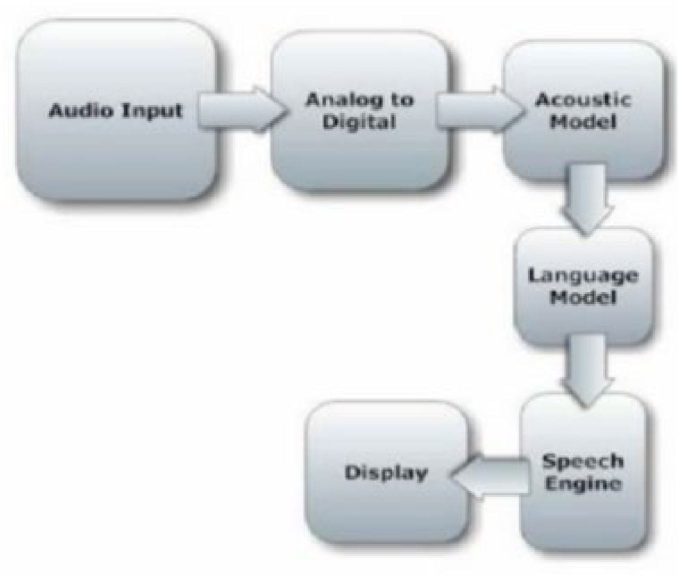


Fig: 2.3 Speech Recognition Process

2.4 APPLICATION

I. FROM MEDICAL PERSPECTIVE
II. FROM MILITARY PERSPECTIVE
III. FROM EDUCATIONAL PERSPCTIVE
IV. FROM COMMERCIAL PERSPECTIVE

CHAPTER 3

(SYSTEM DEVELOPMENT)

3.1 Speech Synthesis

3.1.1 Evaluation of Synthetic Speech:

Speech Synthesis Systems can be calculate I terms of different requirementssuch as speech intelligibility, Speech Naturalness, System Complexity, and so on.For Ambient Intelligent Application it is Reasonable to imagine that new Evaluation Criteria will be Require for example , emotional Influence on the User, Ability to get the User to Act, mastery over Language generation, and Whether the system takes the Environmental Variables into Account and adjusts its behaviour Accordingly

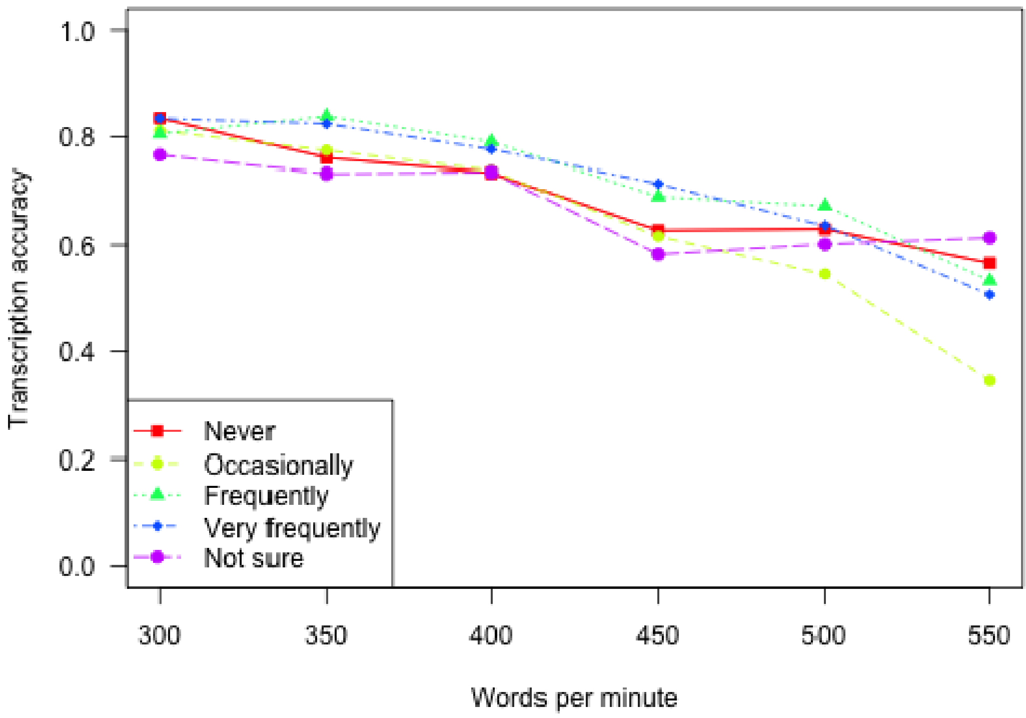
Some Of the Just Mentioned evaluation Criteria are for the Complete System . Having Evaluation Criteria for the Whole System is reasonable because a single, misperforming component would negatively impact how the system is perceived by humans.

3.1.2 Building Speech Synthesis Systems:

Building Speech Synthesis Systems require a speech Units Corpus. Natural Speech must have been recorded for all Units- For Example, all Phonemes – in all possible Contexts.

Next the Units in the Spoken Speech Data are segmented and labelled. Finally, the most Appropriate Speech Units are Chosen (Black and Campbell, 1995).

Generally, concatenative Synthesis yields high quality Speech. With the Large Speech Units Corpus, high quality speech waveforms can be generated. Such synthesized speech preserves waveforms can be generated. Such synthesised speech preserves naturalness and intelligibility. Separate prosody modelling is not necessary for speech unit selection due to the availability of many units corresponding to varied contexts.



GRAPH 3.1

(Line graph showing transcription accuracy by speaking rate for expert and non-expert users of text-to-speech synthesizers)

3.2 Packages Used :

The following is the install packages in this project:

1. import speech_recognition **:** speech_recognition helps to take the input with ease and helps in running model in just a few minutes.

   The speech_recognition library has several popular speech APIs and is thus extremely flexible. It consists of seven APIs which can be used to speech recognition but all six APIs comes with authentication key and password except Google_speech API which makes it extremely flexible and with its ability of free usage and ease of use it makes it excellent choice for speech recognition.

2. import pyaudio: The pip install pyAudio command installs pyaudio to the python interpreter and thus make it easier to work with microphones which helps in real time speech recognition.
   With PyAudio, we can easily use Python to record and to play audio on a kind of variety of platforms.

3. import web browser: with this package we can make use of our default browser used to locate, retrieve and display data .The URL and the query is passed to the instance of the webbrowser package and basis on the url provided and the query the particular webpage opens.

Recognizer Class : All the major process for speech recognition occurs in the recognizer class. As the main function or purpose of a recognizer instance is that to recognize speech and it provides with the various processes and functions which furthur helps in recognising speech from audio source.

Each Recognizer instance is having 7 methods for recognizing speech from the audio source with using various APIs. These are:

- recognize_bing(): Microsoft Bing Speech
- recognize_google():Google Web Speech API
- recognize_google_cloud(): Google Cloud Speech - requires installation of the google-cloud-speech package
- recognize_houndify(): Houndify by SoundHound
- recognize_ibm(): IBM Speech to Text
- recognize_sphinx(): CMU Sphinx - requires installing PocketSphinx
- recognize_wit(): Wit.ai

The 6 APIs require authentication of either an API key or an username/password combination.Therefore we have used the google's web speech of API in this project.

3.2.1 Working With Audio Files:

After the installation of Speech Recognition in the command line it becomes easy to use the audio files because of its Audiofile class.

The path of the audio file can be passed as the argument to the AudioFile class and it also provides with the context manager as it helps in reading and working with the file material.

The context manager then is responsible for opening of the audio file and finally stores the data of file in the instance of the AudioFile.Then the record() method is used to store the data from the entire audio file and initialize it into the instance of AudioData.

The recognize google() is used to recognize any kind of speech in the audio. The results depends on the internet's connection speed and are displayed and the speech to text conversion depends immensly on the accent and the speed of the speaker.As we have used the audio file our speech recognition system caught some words differently because of the vocabulary of the speaker.

Implementation of Audio Files Working:

```
import speech_recognition as sr
r = sr.Recognizer()



harvard = sr.AudioFile('harvard.wav')
with harvard as source:
    audio = r.record(source)
    audio1 = r.record(source,offset=4,duration=5)

print('\n\n\n')
print('Audio file Contents printed')
r.recognize_google(audio)


print('\n\n\n')
print('Audio file content printed while offset and duration is used')
print('\n\n\n')
r.recognize_google(audio1)
```

Output For the audio files:

```
Console 1/A
    ...:
    ...:
    ...: print('\n\n\n')
    ...: print('Audio file Contents printed')
    ...: r.recognize_google(audio)



Audio file Contents printed
Out[13]: 'the still smell of old beer drinkers it takes hi to bring out the order I
called it yourself invest a salt a kotess find the M tacos Al pastor my favourite is
just for food is Bihar cross bun'

In [14]:
```

The offset and duration keywords are useful for modifications in the audio. The offset argument in the record method tells about the starting point and duration tells about the time upto which the conversion is to be made. E.g: if offset value = 5 then the audio file is trimmed to the first five seconds and then rest of speech is used. If duration =5 then the audio file speech is converted for five seconds only and rest of audio is not recoginized.

```
Console 1/A
    ...: #print('Audio file contents printed')
    ...: #r.recognize_google(audio)
    ...:
    ...:
    ...: print('\n\n\n')
    ...: print('Audio file content printed while offset and duration is used')
    ...: print('\n\n\n')
    ...: r.recognize_google(audio1)



Audio file content printed while offset and duration is used



Out[14]: 'ethics hi to bring out the order I called it restore it help'
```

Fig Usage of Offset and duration

3.2.2 The Effect of Noice on Speech Recognition

All of the audio recordings consists of some level of noise in them & the unhandled noise can greatly reduce the accuracy of the speech recognition apps.

This file has the phrase “smell during periods” spoken with a loud sound in the background. Thus the speech cannot be recognised properly.

Input For The noisy audio file:

```
''' Working with noisy input files'''


jackhammer = sr.AudioFile('jackhammer.wav')
with jackhammer as source:
  audio2 = r.record(source)


print('\n')
print('Audio file with background noise displays wrong text')
print('\n\n')
r.recognize_google(audio2)


''' Noise removal to improve speech quality'''

with jackhammer as source:
  r.adjust_for_ambient_noise(source, duration=1)
  audio3 = r.record(source)

print('\n')
print('removing the background noise !!displays:')
print('\n\n')
r.recognize_google(audio3)
```

Output For The noisy audio file:

Console 1/A

```
In [17]: with jackhammer as source:
    ...:   r.adjust_for_ambient_noise(source, duration=0.5)
    ...:   audio3 = r.record(source)
    ...:
    ...:
    ...: print('\n')
    ...: print('removing the background noise !!displays:')
    ...: print('\n\n')
    ...: r.recognize_google(audio3)


removing the background noise !!displays:



Out[17]: 'smell during periods'
```

3.3.3 Speech Recognition Using Microphone:

The Pyaudio is installed to access the microphone which helps the user for real -time speech recognition.With instance of speech_recognition the microphone can be used.

Input Through the microphone:

```
'''for index, name in enumerate(sr.Microphone.list_microphone_names()):
    print("Microphone with name \"{1}\" found for `Microphone(device_index={0})`".format(index, name))

with sr.Microphone() as source:
   print("Analyze the source for 5 seconds")
   r.adjust_for_ambient_noise(source, duration=5)
   print('say something')
   audio1 = r.listen(source)


try:

        print("Google Speech Recognition thinks you said " + r.recognize_google(audio1))

except sr.UnknownValueError:
        print("Google Speech Recognition could not understand audio")

except sr.RequestError as e:
        print("Could not request results from Google Speech Recognition service; {0}".format(e))
```

Output of the Speech input taken from the microphone:

```
In [3]: runfile('C:/Users/Ashish/Documents/Major Project Speech/speech.py', wdir='C:/
Users/Ashish/Documents/Major Project Speech')
Microphone with name "Microsoft Sound Mapper - Input" found for
`Microphone(device_index=0)`
Microphone with name "Microphone Array (Realtek High " found for
`Microphone(device_index=1)`
Microphone with name "Microsoft Sound Mapper - Output" found for
`Microphone(device_index=2)`
Microphone with name "Speakers (Realtek High Definiti" found for
`Microphone(device_index=3)`
Analyze the source for 5 seconds
say something
Google Speech Recognition thinks you said speech recognition system major project

In [4]:
```

3.2.4 Guess The Fruit Game:

Using the speech Recognition we implemented a game of guessing a word with the help of the input provided by the microphone of the user.In this game the user is provided with the list of fruits names and the number of attempts required by the user to guess the fruit which is being guessed by the speech recognition system.

If the user guesses the fruit name correctly then the game announces the win else it prints the message to try again if their are any attempts remaining.

3.2.4.1 Working :

The function recognize_speech_mic() takes two arguments, recognizer and microphone and return a dictionary with three keys. The first key

,success is of type bool which tells about the request made to the API. The second key error is used because it returns None and error message if the API is unavailable and speecch was not recognised. The last key trancription contains the audio recorded by the microphone.

This function first checks the correctness of both the arguments and produces a Type Error if anyone of them is invalid.

Then listen method is used to liste to the input from the mic.

The adjust_for_noise method is used to change the noise conditons each time the function recognise is called which provides clear transcription of the input speech to the user.

Then the recognise_google is called to transcribe the speech from the recording.A try and ecept block is used to catch the Request Error and Unknown ValueError and are handled by returning the resonse.

```
# adjust the recognizer sensitivity to ambient noise and record audio
# from the microphone
with microphone as source:
    recognizer.adjust_for_ambient_noise(source)
    audio = recognizer.listen(source)

# set up the response object
response = {
    "success": True,
    "error": None,
    "transcription": None
}


try:
    response["transcription"] = recognizer.recognize_google(audio)
except sr.RequestError:
    # API was unreachable or unresponsive
    response["success"] = False
    response["error"] = "API unavailable"
except sr.UnknownValueError:
    # speech was unintelligible
    response["error"] = "Unable to recognize speech"

return response
```

The response dictionary is responsible for returning the success of API request ,any error messages and the transcribed speech and the values of each key is stored accordingly which is returned from the recognize_speech_mic function.

The game is quite simple ,we first declare the list of fruits and number of gusses then we create the instance for the Recognizer and Microphone and random word is choosen from the list of the fruits.

```
if __name__ == "__main__":
    # set the list of words, maxnumber of guesses
    WORDS = ["apple", "banana", "grape", "orange", "mango", "lemon"]
    NUM_GUESSES = 3
    PROMPT_LIMIT = 5

    # create recognizer and mic instances
    recognizer = sr.Recognizer()
    microphone = sr.Microphone()

    # get a random word from the list
    word = random.choice(WORDS)

    # format the instructions string
    instructions = (
        "I'm thinking of one of these words:\n"
        "{words}\n"
        "You have {n} tries to guess which one.\n"
    ).format(words=', '.join(WORDS), n=NUM_GUESSES)
```

```python
    print(instructions)
    time.sleep(3)

    for i in range(NUM_GUESSES):

        for j in range(PROMPT_LIMIT):
            print('Guess {}. Speak!'.format(i+1))
            guess = recognize_speech_from_mic(recognizer, microphone)
            if guess["transcription"]:
                break
            if not guess["success"]:
                break
            print("I didn't catch that. What did you say?\n")

        if guess["error"]:
            print("ERROR: {}".format(guess["error"]))
            break

        print("You said: {}".format(guess["transcription"]))

        guess_is_correct = guess["transcription"].lower() == word.lower()
        user_has_more_attempts = i < NUM_GUESSES - 1


        if guess_is_correct:
            print("Correct! You win!".format(word))
            break
        elif user_has_more_attempts:
            print("Incorrect. Try again.\n")
        else:
            print("Sorry, you lose!\nI was thinking of '{}'.".format(word))
            break
```

Then the instructions are printed which tells the user that the speech Recogition system is thinking of one word and the number of guess given to the user.After that the sleep(n) function wait for n seconds.

The first for loop of the program runs for the number of guesses provided to the user.the othe for loop inside the first for loop attempts to recognise the input each time from the recognise_speech_mic() function which stores the dictonary returned from this function and stores it in an variable

If the system recognises the word spoken by the user .I.e the transcription key is not null and the speech of user is transcribed and the inner loop breaks out and if the speech is not transcribed and the API error occured then also the loop breaks out and if the API request becomes succesful but the speech was not recognised then the else statement is executed which tells the user to again speak the word.

If the inner loop breaks out without any errors then the returnedd dicitionary is correct the errors if the error occured then the error message is displayed and which ends the program.

If no error occur on the breaking of the inner loop then the inscription is cmatched is compared to the word selected by system and to lower() method is used to convert string into lowercases which reduces the possiblity of wrong answer because of the conversion of the speech to upper cases .

If user makes a correct guess which matches with the system’s guess then the user win the game else the outer loop executes on the basis of the attempts left and finally if user fails in last attempt then user loses the game.

<u>Output For Guess The Fruit Using Speech Recognition Through Microphone:</u>

IPython console

Console 1/A

```
In [21]:

In [21]: runfile('C:/Users/Ashish/Documents/Major Project Speech/game.py', wdir='C:/
Users/Ashish/Documents/Major Project Speech')
I'm thinking of one of these words:
apple, banana, grape, orange, mango, lemon
You have 3 tries to guess which one.

Guess 1. Speak!
You said: hello
Incorrect. Try again.

Guess 2. Speak!
I didn't catch that. What did you say?

Guess 2. Speak!
I didn't catch that. What did you say?

Guess 2. Speak!
You said: Apple
Incorrect. Try again.

Guess 3. Speak!
I didn't catch that. What did you say?

Guess 3. Speak!
You said: mango
Sorry, you lose!
I was thinking of 'orange'.

In [22]:
```

3.3 System Development Approach :

3.3.1 ACTIVITY DIAGRAM:

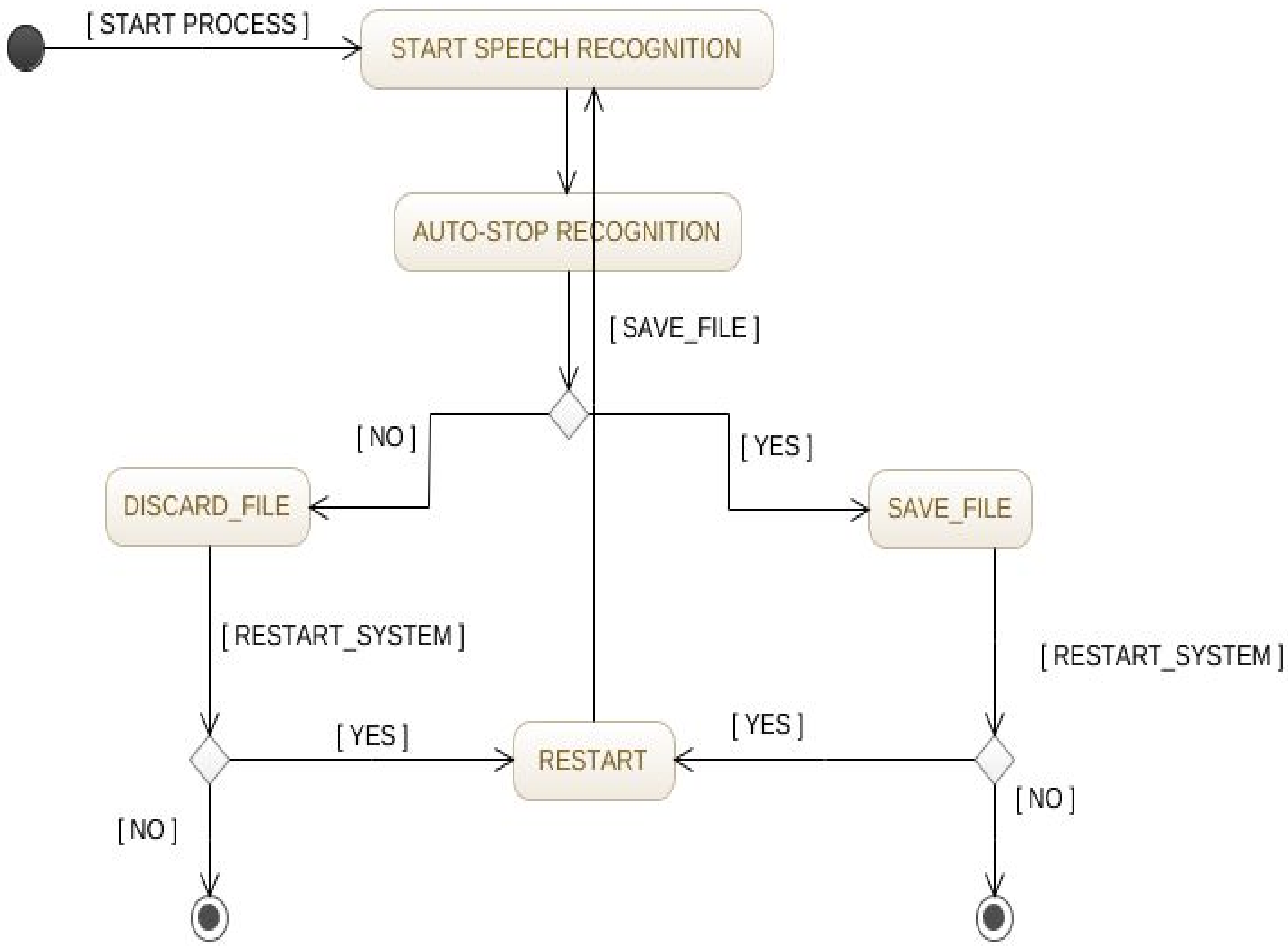


Fig: 3.1 ACTIVITY DIAGRAM

3.3.2 CLASS DIAGRAM:

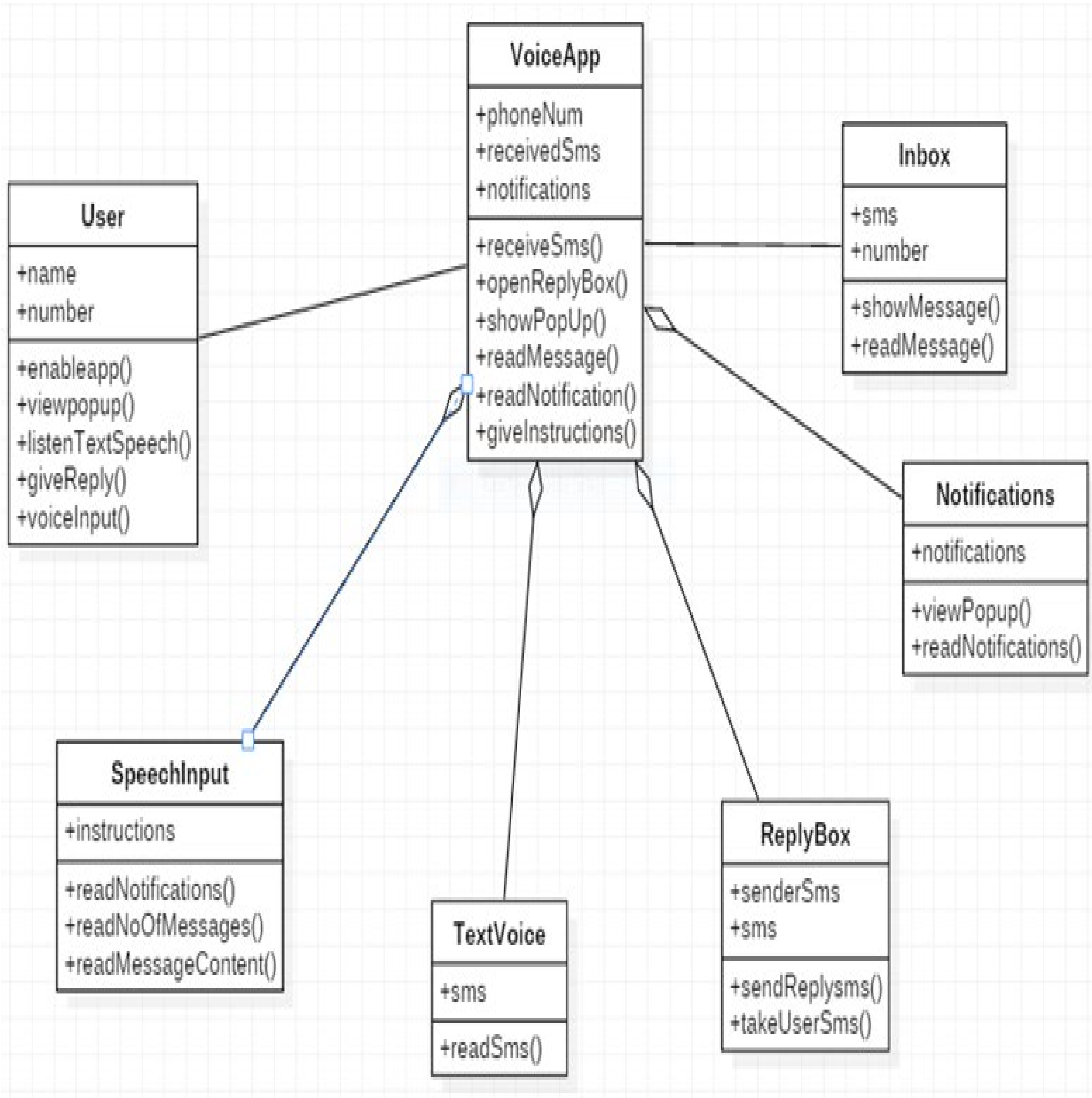


Fig: 3.2 CLASS DIAGRAM

3.3.4 SEQUENCE DIAGRAM:

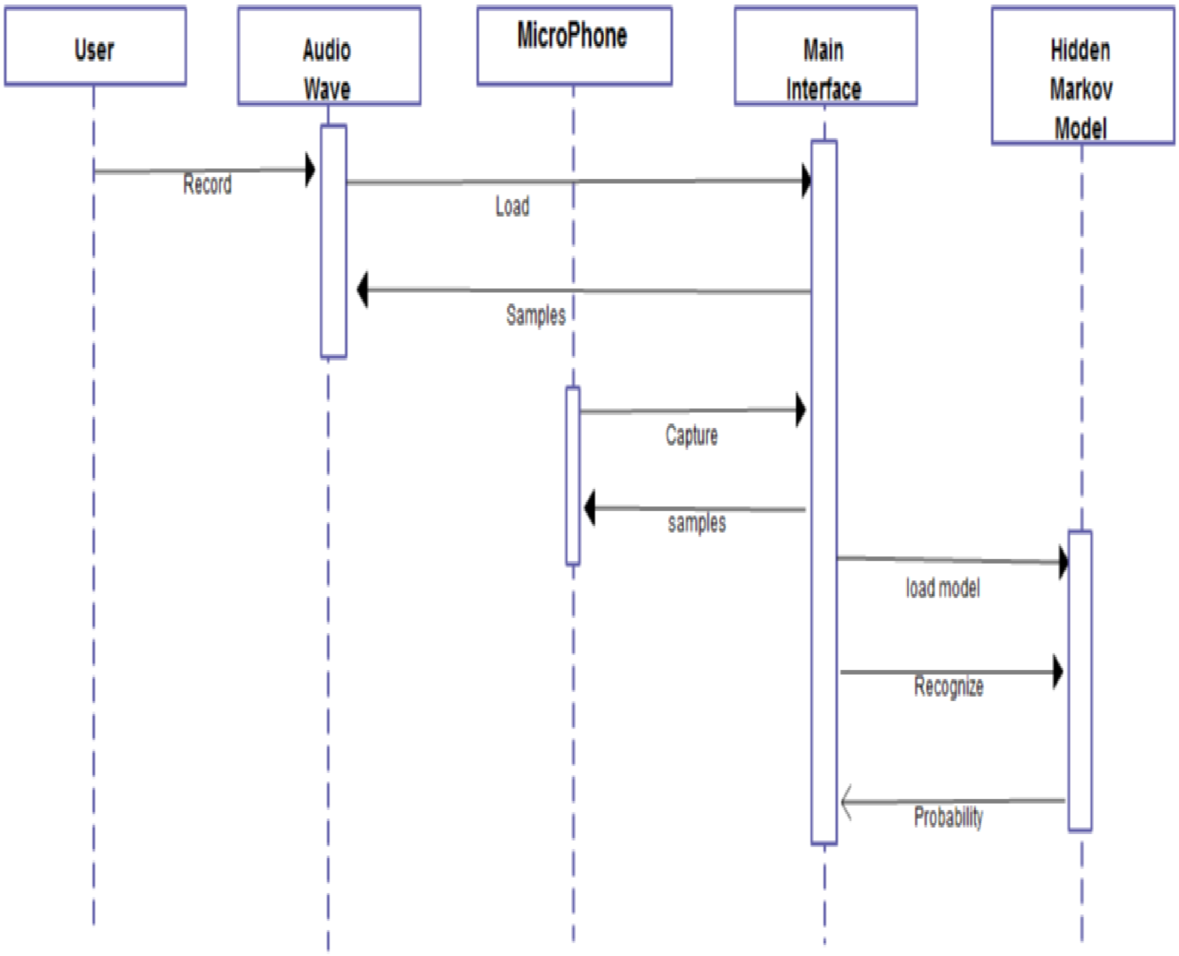


Fig: 3.4 SEQUENCE DIAGRAM

CHAPTER 4

(PERFORMANCE ANALYSIS)

4.1 System Requirement:

4.1.1 Requirements:

a. 1.6 MHz Processor

b. 128 MB RAM

c. Microphones for good audio.

4.1.2 Best Requirements:

a. 2.4 GHz processor

b. Greater than128 MB RAM

c. 10% consumption of memory

d. best quality microphones

4.2 Hardware Requirement

Sound cards:

The proper driver must be installed for the sound as speech requires low Bandwidth thus high quality of sound cards are to be used.

Microphones:

Microphones are the most important tools for the real time speech to text conversion .Therefore the pre-installed ones cannot be used as they are more prone to the background noise and also of poor quality in terms of speech.

Computer Processor:

Speech recognition application depends majorly on processing speed. The input from the user can take some time if the processing speed is low and thus user wasted more time on waiting compared to performing the task which makes the application less feasible for use.

4.3 Web Search Using Speech Recognition:

We will make a program using the speech Recognition python to execute the following:

1. Conversion of speech to text.
2. Using the text to open a URL using web browser
3. Searching a query using speech inside the URL.

The program imports speech recognition library which handles the request from the user to perform web search and search the query on the youtube.

For performing web search we used the Recognizer class of the speech recognition and created three instances of this class.

first instance is used to recognize text from youtube ,second instance is used for web search and third instance is used to listen to speech .

We take input from the user’s microphone and on the basis of the words spoken e.g: web search and video we search the web and youtube respectively.

The microphone recognizes the speech using recognize_google() method and using listen method we record the input from the source and outputs the web browser page.

This system is designed to recognize the speech and also has the capabilities to convert speech to text. This software name ‘SPEECH RECOGNITION SYSTEM’ has the capability to write spoken words into text.

Fig: 4.3.1 program for relative searches

```python
with sr.Microphone() as source:
  print('[search python : search Youtube]')
  print('speak Now!! \n')
  audio = r3.listen(source)


if 'python' in r2.recognize_google(audio):
    r2 = sr.Recognizer()
    url = 'https://www.edureka.co/'
    with sr.Microphone() as source:
      print('\n search the query \n')
      audio = r2.listen(source)

    try:
      get = r2.recognize_google(audio)
      print(get)
      wb.get().open_new(url+get)
    except sr.UnknownValueError:
      print('Unable to recognize')
    except sr.RequestError as e:
      print('failed'.format(e))



if 'video' in r1.recognize_google(audio):
    r1 = sr.Recognizer()
    url = 'https://www.youtube.com/results?search_query='
    with sr.Microphone() as source:
      print('\n search the query \n')
      audio = r2.listen(source)


    try:
      get = r1.recognize_google(audio)
      print(get)
      wb.get().open_new(url+get)
    except sr.UnknownValueError:
      print('Unable to recognize')
    except sr.RequestError as e:
      print('failed'.format(e))
```

```
Console 1/A
[search python : search Youtube]
speak Now!!


In [26]: runfile('C:/Users/Ashish/Documents/Major Project Speech/web.py', wdir='C:/
Users/Ashish/Documents/Major Project Speech')
[search python : search Youtube]
speak Now!!


 search the query

free code camp
```

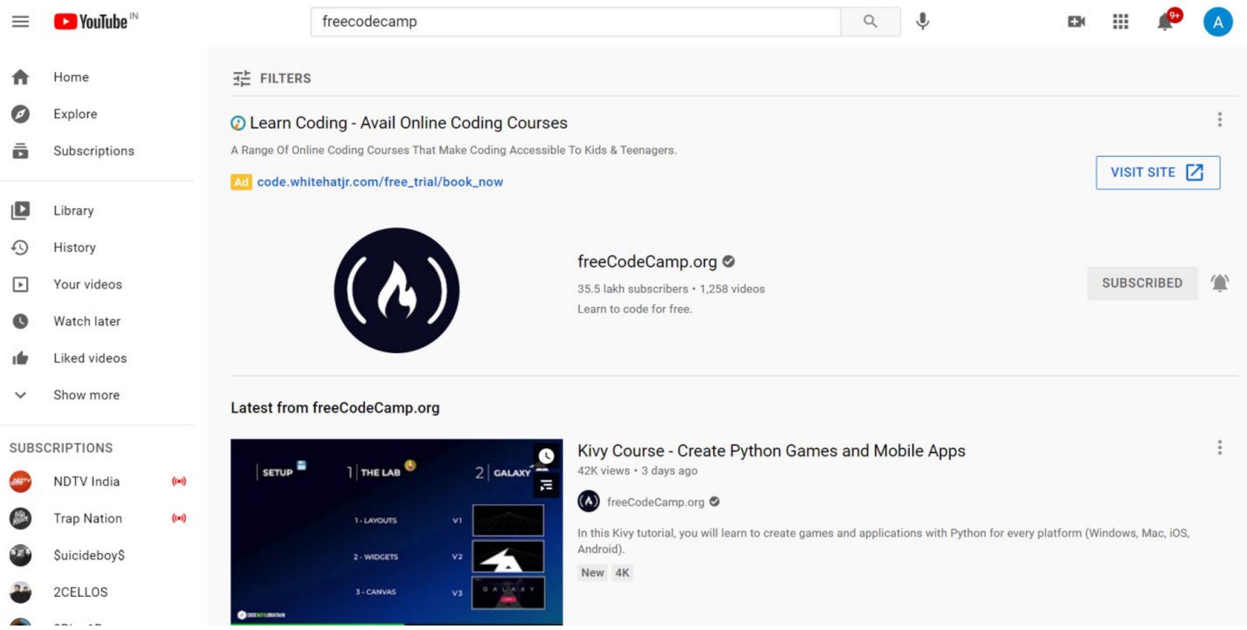
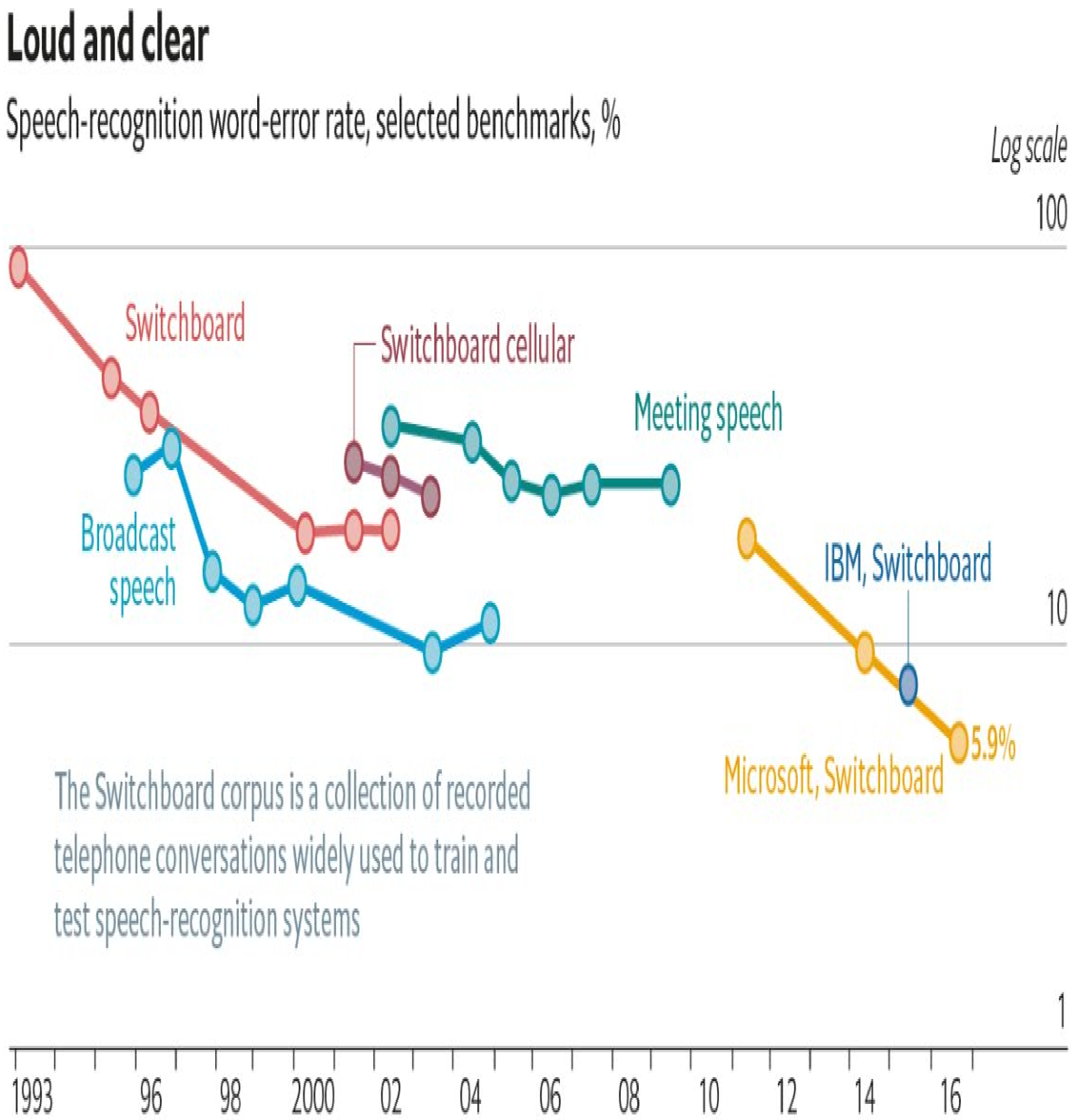


Fig: 4.3.1 Recognize the word freecodecamp

4.1 <u>Graphical Representation:</u>



GRAPH 4.1
(SOURCE: MICRosoft)

CHAPTER 5

(CONCLUSION)

5.1 **Advantages of Software**:

In mostly areas of the country, there are lot of people who don't know how to write and also how to read any word, so this project is very helpful for these type of people as you know in today's world, everybody has its own mobile phones and they want to search a lot of things. In this project, they usually speak what they want to search and various results of such type opens in the browser window.

1. Ability to write text using speech.
2. Different windows can be opened and web searches can be made.
3. More utilization of resources and less time consumption.
4. Recognises different audio files and convert them to text.
5. Helpful for disabled peoples.

5.2 **Disadvantages:**

1. Low accuracy because of its limited ability.
2. Fails in noisy environment.
3. Depends majorly on GoogleAPI thus not a original software.
4. Limited operations can be performed.

5.3 Conclusion:

The project of speech recognition gives us the introduction of this technology and its various application in different sectors. The project is divided into three parts ,the first which helps in converting audio to text ,the second which recognises the spoken word and the third which performs the operations provided as the command by the user.After the development of these parts these models were tested and the results were produced which tells about the accuracy of each model.Various advantages and disadvantages of this software is discussed.

REFERENCES

BOOKS:

1. G. L. Clapper, "Automatic word recognition", IEEE Spectrum, pp. 57-59, Aug. 1971.

View Article Full Text: PDF (8868KB) Google Scholar

2. M. B. Herscher, "Real-time interactive speech technology at threshold technology", Workshop Voice Technol. Interactive Real Time Command Control Syst. Appl., 1977-Dec.

Google Scholar

3. J. W. Gleen, "Template estimation for word recognition", Proc. Conf. Pattern Recog. Image Processing, pp. 514-516, 1978-June.

Google Scholar

Internet:

1. https://pypi.org/project/SpeechRecognition.

2. https://www.researchgate.net/publication/337155654_A_Study_on_Automatic_Speech_Recognition.

3. https://www.ijedr.org/papers/IJEDR1404035.pdf

Rahul Tomar

ORIGINALITY REPORT

29%	22%	7%	16%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	sonix.ai Internet Source	6%
2	Submitted to IUBH - Internationale Hochschule Bad Honnef-Bonn Student Paper	6%
3	realpython.com Internet Source	6%
4	S. Jothilakshmi, V.N. Gudivada. "Large Scale Data Enabled Evolution of Spoken Language Research and Applications", Elsevier BV, 2016 Publication	2%
5	Murtaza Bulut. "Speech Synthesis Systems in Ambient Intelligence Environments", Human-Centric Interfaces for Ambient Intelligence, 2010 Publication	2%
6	towardsdatascience.com Internet Source	2%
7	www.hindawi.com Internet Source	1%

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT

PLAGIARISM VERIFICATION REPORT

Date: ……………………….

Type of Document (Tick): **PhD Thesis** **M.Tech Dissertation/ Report** **B.Tech Project Report** **Paper**

Name: Akshit Chaudhary and Rahul Tomar **Department:** CSE/IT **Enrolment No** 171295 ,171271

Contact No. ______________________ **E-mail.** 171295@juitsolan.in and 171271@juitsolan.in

Name of the Supervisor: Dr. Pradeep Kumar

Title of the Thesis/Dissertation/Project Report/Paper (In Capital letters): ______________

SPEECH RECOGNITION USING PYTHON

UNDERTAKING

I undertake that I am aware of the plagiarism related norms/ regulations, if I found guilty of any plagiarism and copyright violations in the above thesis/report even after award of degree, the University reserves the rights to withdraw/revoke my degree/report. Kindly allow me to avail Plagiarism verification report for the document mentioned above.

Complete Thesis/Report Pages Detail:

- Total No. of Pages = 39
- Total No. of Preliminary pages = 37
- Total No. of pages accommodate bibliography/references = 1

(Signature of Student)

FOR DEPARTMENT USE

We have checked the thesis/report as per norms and found **Similarity Index** at ……29…………(%). Therefore, we are forwarding the complete thesis/report for final plagiarism check. The plagiarism verification report may be handed over to the candidate.

(Signature of Guide/Supervisor) **Signature of HOD**

FOR LRC USE

The above document was scanned for plagiarism check. The outcome of the same is reported below:

Copy Received on	Excluded	Similarity Index (%)	Generated Plagiarism Report Details (Title, Abstract & Chapters)	
	• All Preliminary Pages • Bibliography/Images/Quotes • 14 Words String		Word Counts	
Report Generated on			Character Counts	
		Submission ID	Total Pages Scanned	
			File Size	

Checked by
Name & Signature **Librarian**

……………………………………………………………………………………………………

Please send your complete thesis/report in (PDF) with Title Page, Abstract and Chapters in (Word File) through the supervisor at plagcheck.juit@gmail.com