



Report On

DESKTOP VOICE ASSISTANT

For Course

Artificial Intelligence

(INT 404)

BY:

Registration No	Name of Student	Roll No	Section	Project Group No
11804437	Rishabh Gupta	14	K18SP	07
11804429	Satyam Trivedi	15	K18SP	07
11804431	Vaibhav Patel	24	K18SP	07
11804445	Rajat Tiwari	25	K18SP	07

Submitted To: **Manu Bali**

Faculty of: **Lovely Professional University**

Jalandhar, Punjab, India

PROJECT OVERVIEW

This thesis report considers an overview of voice recognition technology, software development, and its applications. The description of voice recognition process, its applications in different sectors, its flaws and finally the future of technology. Voice recognition is commonly used to operate a device, perform commands, or write without having to use a keyboard, mouse, or press any buttons, software program has ability to decode the human voice. Basically first automatic speech recognition(ASR) or voice recognition was used in 1952 which recognized single digits spoken by user because it was not computer driven but today this technology is using everywhere like industries, healthcare, military, telecommunication and personal computing.

There are plenty daily life uses :-

Automated phone systems - Many companies today use phone systems that help direct the caller to the correct department. If you have ever been asked something like "Say or press number 2 for support" and you say "two," you used voice recognition.

Google Voice - Google voice is a service that allows you to search and ask questions on your computer, tablet, and phone.

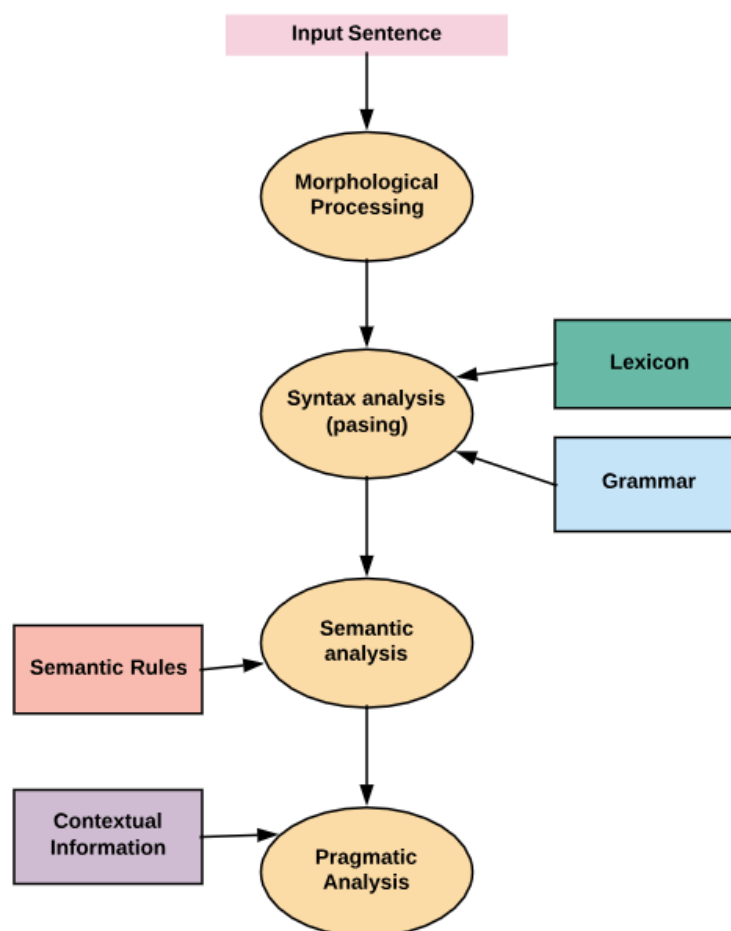
Digital assistant - Amazon Echo, Apple's Siri, and Google Assistant use voice recognition to interact with digital assistants that helps answer questions.

Atlast this process have been in a improving state by big companies like google, microsoft. Later part covers the speech recognition process, and the code for the software and its working. Finally the report concludes at the different potentials uses of the application and further improvements and considerations, this project is working on Natural Language Processing (NLP) and speech recognition which makes interaction easy and interactive between human and machines (Systems).

NATURAL LANGUAGE PROCESSING

Natural Language Processing or NLP is a field of Artificial Intelligence that gives the machines the ability to read, understand and derive meaning from human languages.

It is a discipline that focuses on the interaction between data science and human language, and is scaling to lots of industries. Today NLP is booming thanks to the huge improvements in the access to data and the increase in computational power, which are allowing practitioners to achieve meaningful results in areas like healthcare, media, finance and human resources, among others.



Use Cases of NLP

In simple terms, NLP represents the automatic handling of natural human language like speech or text, and although the concept itself is fascinating, the real value behind this technology comes from the use cases.

NLP enables the recognition and **prediction of diseases** based on electronic health records and patient's own speech. This capability is being explored in health conditions that go from cardiovascular diseases to depression and even schizophrenia. For example, Amazon Comprehend Medical is a service that uses NLP to **extract disease conditions**, medications and treatment outcomes from patient notes, clinical trial reports and other electronic health records. NLP is particularly booming in the **healthcare industry**. This technology is improving care delivery, disease diagnosis and bringing costs down while healthcare organizations are going through a growing adoption of electronic health records. The fact that clinical documentation can be improved means that patients can be better understood and benefited through better healthcare. The goal should be to optimize their experience, and several organizations are already working on this.

VOICE RECOGNITION

Voice recognition technology is concerned with using voice samples to identify a speaker (voice identification) or to confirm the identity of a speaker (voice verification). These are the two different applications of voice recognition technology.

Digital voice recognition technology works by recording a voice sample of a person's speech and digitizing it to create a unique voice print or template. Each spoken word is broken up into discrete segments which

comprise several tones. These tones can be digitized and are captured to create a speaker's unique voice template.

Physiological component:

The physiological component of a person's voice is based on the shape of that person's vocal tract, i.e. the shape of the larynx, nose and mouth. Biometric speech technology uses the wave form of the voice sample to digitally recreate the shape of an individual's vocal tract. No two individuals can have the same vocal tract and therefore every person will have a unique voice imprint.

Behavioral component:

This component represents the physical movement of the individual's jaw, tongue and larynx. Variation in this movement causes changes in the pace, manner and pronunciation of a person's voice, which include the person's accent, tone, pitch, pace of talking etc.

Factors that can cause a template mismatch:

Currently voice recognition systems are not completely infallible, and there are factors that can cause a voice recognition system to return an erroneous result.

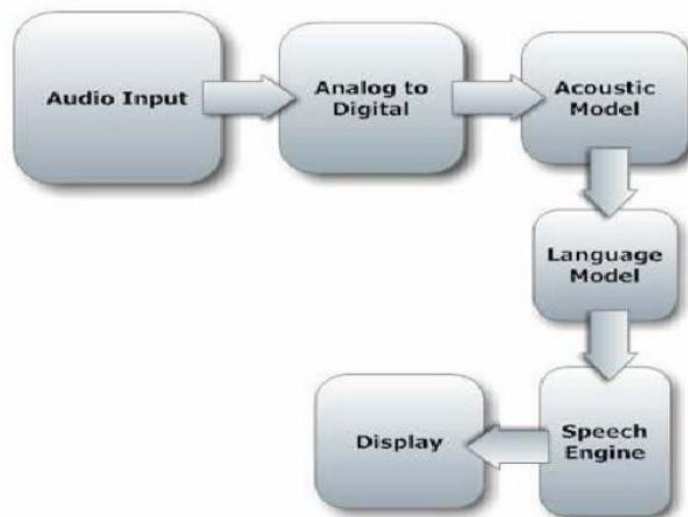
Differences in the physical well being and emotional state of an individual can also cause a person's speech to change. For example, having a cold can change the person's speech enough to create a mismatch between the stored template and the person's current voice sample. This is even possible if the person is excited, depressed, stressed or even on medication. Other factors, such as background noise, ambient temperature and the input device can also affect the result of the system.

There are two main types of voice recognition systems

Text dependent: These require the speaker to say a predetermined word / phrase, which is called a “pass phrase”. The pass phrase can be anything, ranging from the name of the person, their birthplace to their favorite color or a set of numbers. The phrase is compared to a sample captured during enrollment.

Text independent: These systems are trained to recognize a person without a pass phrase. However, they require longer speech inputs from the speaker in order to identify distinct vocal characteristics such as pitch, cadence and tone. Some systems incorporate both text dependent and independent systems. Owing to the countless benefits that voice recognition systems have, they are becoming more popular with each passing day.

Speech Recognition Process



Project scope

This project has the voice recognizing and speech synthesizing capabilities. This Voice recognizing software is generally based for the user who can depend on doing the regular works such as sending emails, opening internet as well as using their systems with their voice command like playing movies, songs, writing texts opening any kind of software's on the device.

This project is generally user friendly and the user can change the code and design according to the command he wants to add as well as delete. This software can be run even in the cell phone devices and as well as the computers, so this software is reliable.

This project is completely based for the processing of the user by which he can easily do the work and manages the times and this software is completely dependable and as well as user modification facility is added.

Project Objective

- To understand the speech recognition and its fundamentals.
- Its working and applications in different areas
- Its implementation as a desktop Application
- Development for software that can mainly be used for:
 - Speech Recognition
 - Speech Generation
 - Text Editing
 - Tool for operating Machine through voice
- Make interaction between human and Machine much simpler and smarter.

Algorithm

Step-1: Initialize the variable and data which is required to the program and also set the voice of the system.

Step-2: Define a function named speak which give the output to the user.

Step-3: Define another function which wish to the user when program is start according to the time.

Step-4: Define another function which named as takecomand which take input to the user.

Step-5: Repeat step -6 to step -8 while True:

Step-6: Take as by calling step-4.

Step-7: If input query is the data then execute otherwise nothing.

Step-8: If stop query in the input then break/stop the execution.

WORKING OF PROJECT

- In this project generally the voice command is used for mailing the information the user wants to send the dedicated person. According to the user, the user can add any amount of mails so that the work becomes ease to do.
- We can use this project in our workplace into incorporating simple tasks to increase efficiency, as well as beyond tasks that have traditionally needed humans, to be performed.
- We can use this mechanism for searching the documents in the system. We can imply for finding the processing any kind of softwares the user wants.

- We can use this software for request information regarding your balance, transactions, and spending habits without having to open your cell phone.
- First we have import certain libraries like date and time, Wikipedia, web browser etc. Which will used to call that specific functions.
- It can able to send emails to anyone but it's email is available in the database. In future it deals with text messages like WhatsApp or other messaging apps and even phone call also.
- After that, we have initialized variables for further use.
- Moreover, we have used else if command to execute the given command under try catch block.
- Now, why we need to use exception handling in this project, because if the system does not recognized users voice or command, it can simply handle that exception and print "sorry, i can't hear you" message, that will help user to give clear and appropriate command to enjoy this technology.
- This project working on to reduce the effort of the user.

Hardware Requirement

Sound cards:-Speech requires relatively low bandwidth, high quality 16 bit sound card will be better enough to work. Sound must be enabled, and proper driver should be installed. Sound cards with the 'cleanest' A/D (analog to digital) conversions are recommended, but most often the clarity of the digital sample is more dependent on the microphone quality and even more dependent on the environmental noise. Some speech recognition systems might require specific sound cards.

Microphones: A quality microphone is key when utilizing the speech recognition system. Desktop microphones are not suitable to continue with speech recognition system, because they have tendency to pick up more ambient noise. The best choice, and most common is the headset style. It allows the ambient noise to be minimized, while allowing you to have the microphone at the tip of your tongue all the time. Headsets are available without earphones and with earphones (mono or stereo).

Computer/ Processors: Speech recognition applications can be heavily dependent on processing speed. This is because a large amount of digital filtering and signal processing can take place in ASR.

(a). Minimum requirements

Pentium 200 MHz processor

64 MB of RAM

(b). Best requirements

1.6 GHz Processor

128 MB or more of RAM

Work Done in the Project

- Rishabh Gupta: Prepared the report of the project.
- Satyam Trivedi: Implemented the Email functionality.
- Vaibhav Patel: Implemented the Idea in a working model using Python 3.7 Language.
- Rajat Tiwari: Made synopsis and helped Rishabh in doing report work.

OUTPUT VIEW

```

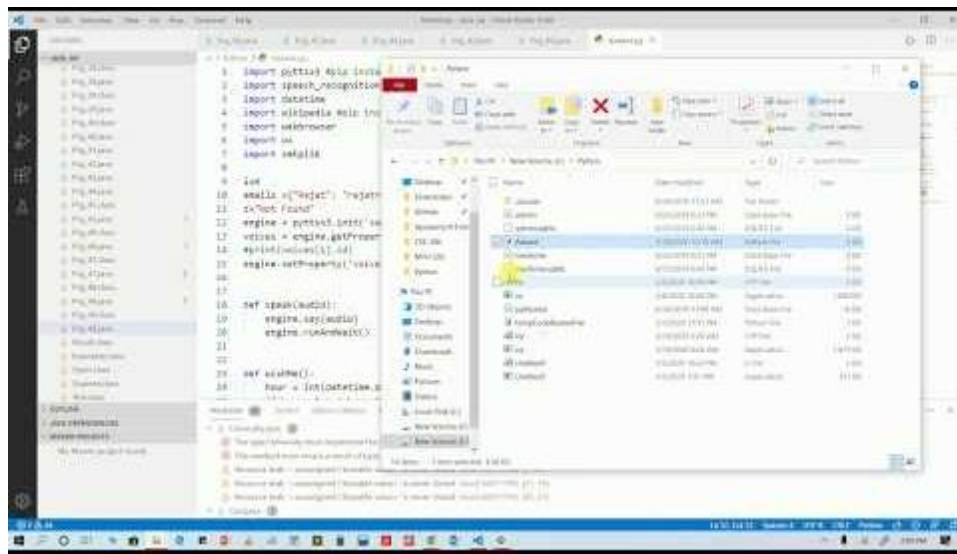
111
112 elif 'stop' in query:
113     speak('stop in 3 2 1 ')
114     break
115
116 elif 'send a ema
117 try:
118     speak("
119     try:
120         per
121         to =
122         if (
123             Listening...
124             Recognizing...
125             Say that again please...
126             Listening...
127             Recognizing...
128             Say that again please...
129             Listening...
130             Recognizing...
131             User said: send email
132             if (
133                 Listening...
134                 Recognizing...
135                 User said: send a email
136                 Listening...
137                 Recognizing...
138                 User said: Vaibhav
139             except B
140             print
141             User said: this is a auto generated email send from our project
  
```

Output View:

```

User said: open u m s
Listening...
Recognizing...
User said: now where the new Browser is open and this is the URL so it is opening the internet connectivity is very slow
so this pages might not be open so I am closing this file
Listening...
Recognizing...
User said: send email
Listening...
Recognizing...
Say that again please...
Listening...
Recognizing...
Say that again please...
Listening...
Recognizing...
User said: send a email
Listening...
Recognizing...
User said: Vaibhav
Listening...
Recognizing...
User said: this is a auto generated email send from our project
  
```

DEMONSTRATION VIDEO OF PROJECT



Link: <https://www.youtube.com/watch?v=hJIQFJRJIk>

CODING OF THE PROJECT

```
import pyttsx3
import speech_recognition as sr
import datetime
import wikipedia
import webbrowser
import os
import smtplib

i=0
emails ={"Rajat": "rajatreigns@gmail.com","Vaibhav": "vniranjan26@gmail.com","Rishabh": "rishabhgupta4999@gmail.com"}
z="Not Found"
engine = pyttsx3.init('sapi5')
voices = engine.getProperty('voices')
engine.setProperty('voice', voices[1].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 Edith my friend. Please tell me how may I help you")

def takeCommand():

    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:
    # 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('aiprojectdemo26@gmail.com', 'Ai@k18sp')
    server.sendmail('aiprojectdemo26@gmail.com', to, content)
    server.close()

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

            if 'wikipedia' in query:
                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")
                speak("youtube is opeing.....")

            elif 'open google' in query:
                webbrowser.open("google.com")
                speak("google is opeing.....")
```

```

elif 'open stackoverflow' in query:
    webbrowser.open("stackoverflow.com")
    speak("stack overflow is opeing.....")

elif 'open u m s' in query:
    webbrowser.open("ums.lpu.in/lpuums/")
    speak("university management system is opeing.....")
)

elif 'play music' in query:
    music_dir = 'D:\Songs'
    songs = os.listdir(music_dir)
    print(songs)
    os.startfile(os.path.join(music_dir, songs[i]))
    i=i+1
    speak("opeing music directory.....")

elif 'what is 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 = "C:\Program Files\Microsoft VS Code\\Code.exe"
    os.startfile(codePath)

elif 'stop' in query:
    speak('stop in 3 2 1...')
    break

elif 'send a email' in query:
    try:
        speak("To whom you want to send email ?")
        try:
            per = takeCommand()
            to = emails.get(per,z)
            if (to==z):
                speak("I didn't have addresss of ")
                speak(per)
                speak("in our data base")
            else :
                speak("What should I say?")
                content = takeCommand()
                sendEmail(to, content)

```

```
        speak("Email has been sent!")
        print("Email has been sent!")

    except Exception as e:
        print(e)

except Exception as e:
    print(e)
    speak("Sorry my friend vaibhav. I am not able to send
this email")
```