

CHAPTER-1 INTRODUCTION

1.1 PURPOSE

The term *voice recognition* or *speaker identification* refers to identifying the speaker, rather than what they are saying. Recognizing the speaker can simplify the task of translating speech in systems that have been trained on a specific person's voice or it can be used to authenticate or verify the identity of a speaker as part of a security process.

From the technology perspective, speech recognition has a long history with several waves of major innovations. Most recently, the field has benefited from advances in deep learning and big data. The advances are evidenced not only by the surge of academic papers published in the field, but more importantly by the worldwide industry adoption of a variety of deep learning methods in designing and deploying speech recognition systems. These speech industry players include Google, Microsoft, IBM, Baidu, Apple, Amazon, Nuance, SoundHound, IflyTek, CDAC many of which have publicized the core technology in their speech recognition systems as being based on deep learning.

1.2 OBJECTIVE

- To build a desktop application that talks to you by use of your voice.
- It's working and functions in different areas.
- Development of software that can be mainly used for tools for operating machine through voice.

1.3 MOTIVATION

Our motivation for creating this application are modern AIs like Siri, Cortana and Google Now.

Siri - Siri is the name of Apple's personal digital assistant. It's basically voice control that talks back to you, that understands relationships and context, and with a personality straight out of Pixar.

Cortana - Cortana is an intelligent personal assistant created by Microsoft for Windows 10, Windows 10 Mobile, Windows Phone 8.1 (where it now supersedes Bing Mobile), Microsoft Band, Xbox One, iOS and Android.

Google Now - Google Now is an intelligent personal assistant developed by Google. Google Now, including Now cards, voice search and commands, is available in the Google app for Android and iOS.

1.4 DEFINITION AND OVERVIEW

- Speech recognition is the inter-disciplinary sub-field of computational linguistics that develops methodologies and technologies that enables the recognition and translation of spoken language into text by computers. It is also known as "automatic speech recognition" (ASR), "computer speech recognition", or just "speech to text" (STT). It incorporates knowledge and research in the linguistics, computer science, and electrical engineering fields.
- Some speech recognition systems require "training" (also called "enrollment") where an individual speaker reads text or isolated vocabulary into the system. The system analyzes the person's specific voice and uses it to fine-tune the recognition of that person's speech, resulting in increased accuracy. Systems that do not use training are called "speaker independent"^[1] systems. Systems that use training are called "speaker dependent".
- Speech recognition applications include voice user interfaces such as voice dialing (e.g. "Call home"), call routing (e.g. "I would like to make a collect call"), domotic appliance control, search (e.g. find a podcast where particular words were spoken), simple data entry (e.g., entering a credit card number), preparation of structured

documents (e.g. a radiology report), speech-to-text processing (e.g., word processors or emails), and aircraft (usually termed Direct Voice Input).

CHAPTER-2 OVERALL DESCRIPTION

2.1 PROJECT PERSPECTIVE

Speech recognition is the ability of a machine or program to identify words and phrases in spoken language and convert them to a machine-readable format. Rudimentary speech recognition software has a limited vocabulary of words and phrases, and it may only identify these if they are spoken very clearly. More sophisticated software has the ability to accept natural speech.

This application will be a developed text recognition system. This will allow user to interact with machine by the use of voice .It will simplify users day to day tasks easier by setting reminder, alarms and other things. It will take voice of user as input and perform corresponding task according to it. It is a kind of intelligent system that will help user to perform various tasks at ease with just a voice command.

2.2 PROJECT FUNCTIONS

- Provide smart replies to general queries
- Web Based Search
- Quick and easy recording sounds and voice memos (notes) and playing it back.
- Simple and friendly operating (handling)
- Renaming your records
- Adding voice reminders to your calendar
- Playing Songs & Videos
- Shut Down
- Restart

2.3 E-R & DFD

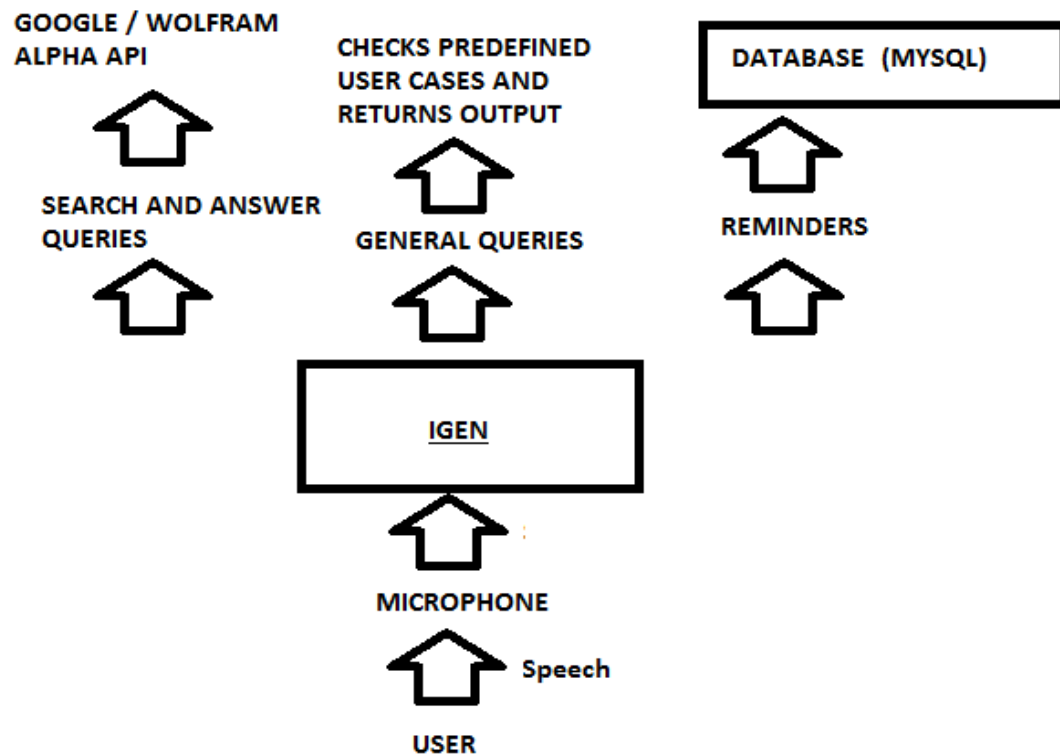


Fig:3.1 Working of DETECTOR

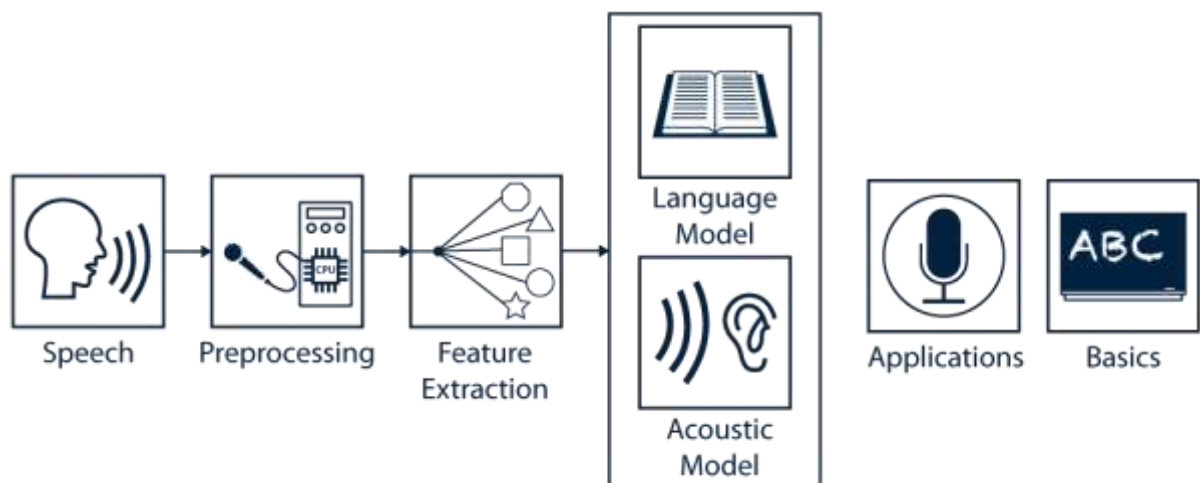


Fig:3.2 Block Diagram of DETECTOR

2.4 CONSTRAINTS AND ASSUMPTIONS

- One of the problems of the speech recognition is not recognizing similar sounding words like ‘here’ and ‘here’, or ‘there’ and ‘their’.
- Only works on Window and has limited functionality.
- If user give a pause in giving input than the input can not be recognized correctly.
- If there is a noise while giving output then the obtain result may not be the one that is expected.
- It is single user application.
- User must know how to operate the computer.

2.4.1 Performance

Speech recognition performance is measured by accuracy and speed. Accuracy is measured with word error rate. WER works at the word level and identifies inaccuracies in transcription, although it cannot identify how the error occurred. Speed is measured with the real-time factor. A variety of factors can affect computer speech recognition performance, including pronunciation, accent, pitch, volume and background noise.

Templates of a same single word	Recognition numbers	Recognition rate
10	5	91.5%
20	5	93.3%
30	5	95.8%

CHAPTER-3 SYSTEM REQUIREMENTS

3.1 EXTERNAL INTERFACE AND REQUIREMENTS

3.1.1 Hardware Interface

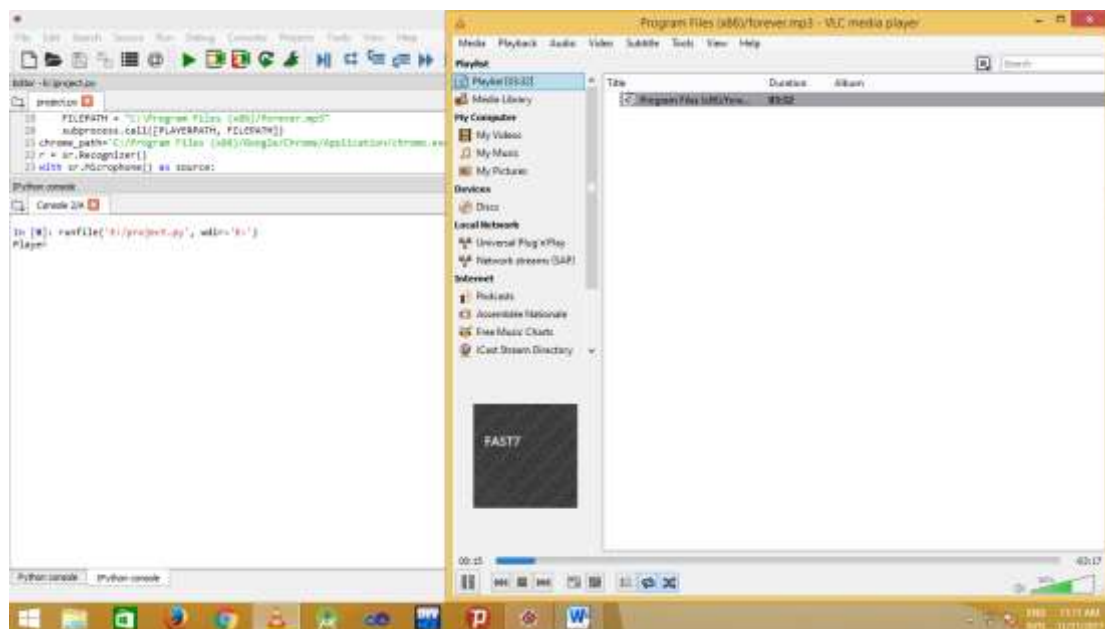
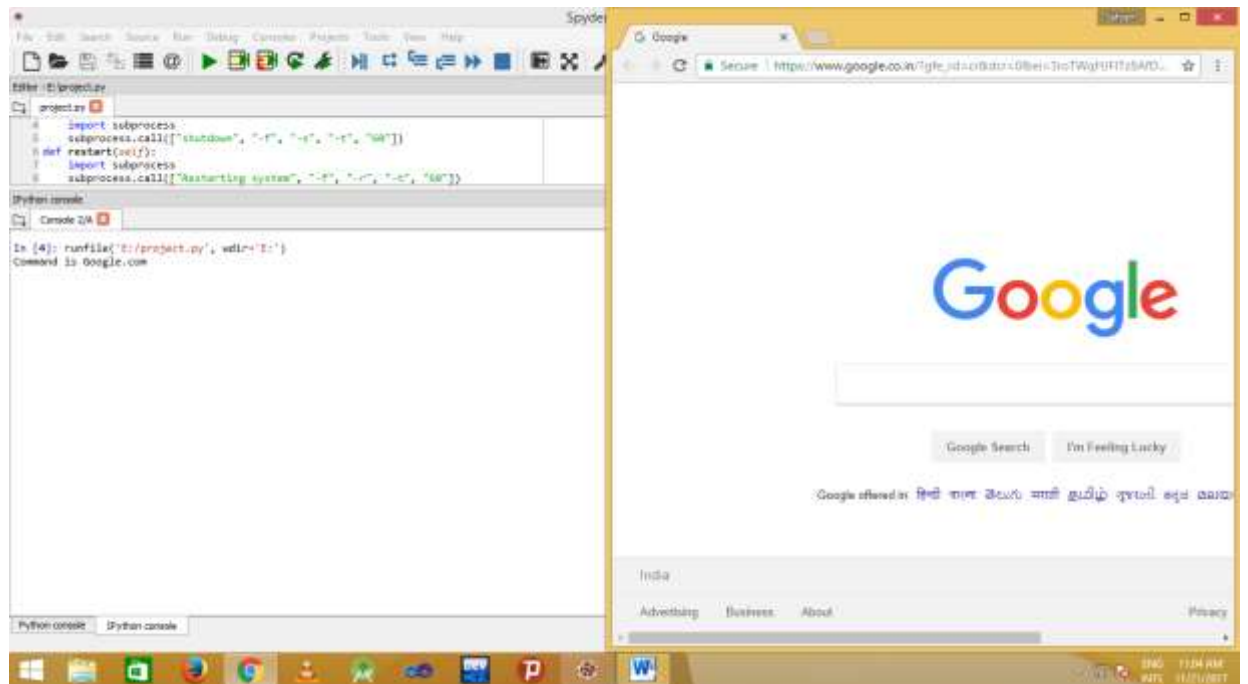
- Microphone



- Speakers



3.1.1 Software Interface



Spyder (Python 3.6)

File Edit Search Source Run Debug Consoles Projects Tools View Help

C:\Users\SATYAN\MITAL

Editor - E:\project.py

project.py

```
1 import speech_recognition as sr
2 import webbrowser as wb
3 def shutdown(self):
4     import subprocess
5     subprocess.call(["shutdown", "-f", "-s", "-t", "60"])
6 def restart(self):
7     import subprocess
8     subprocess.call(["Restarting system", "-f", "-r", "-t", "60"])
9 def logout(self):
10    import subprocess
11    subprocess.call(["Logging Out User", "-f", "-l", "-t", "60"])
12 def abort(self):
13    import subprocess
14    subprocess.call(["Aborting Shutdown", "-f", "-a", "-t", "60"])
15 def player(self):
16    import subprocess
17    PLAYERPATH = "C:/Program Files (x86)/Vidolan/VLC/vlc.exe"
18    print("Player")
19    FILEPATH = "C:/Program Files (x86)/forever.mp3"
20    subprocess.call([PLAYERPATH, FILEPATH])
21 chrome_path="C:/Program Files (x86)/Google/Chrome/Application/chrome.exe %s"
22 r = sr.Recognizer()
23 with sr.Microphone() as source:
24     audio = r.listen(source)
25     user = r.recognize_google(audio)
26
27 if user=="shutdown":
28     shutdown(1)
29 elif user=="restart":
30     restart(1)
31 elif user=="log out":
32     logout(1)
33 elif user=="abort":
34     abort(1)
35 elif user=="player":
36     player(1)
37 else:
38     '''if user[0:3]=="open":
39         ...'''
```

Permissions: RW End-of-lines: CRLF Encoding: ASCII Line: 18 Column: 18 Memory: 40 %

ENG 10:14 AM
NTL 11/21/2017

3.2 FUNCTIONAL REQUIREMENTS

1. Basic Windows Functions

User can perform power operations like shut down, restart, log out and also abort the enlisted Functions.

3. Music

User can also play song stored in his/her computer/laptop.

4. Speech-to-Text

Converts what the user says, performs the function and also shows what the user said.

5. Internet

User can access to Internet facilities by just saying name of website and it will automatically open the same in web browser.

3.3 NON-FUNCTIONAL REQUIREMENTS

- Accessibility: - Can be accessible only by single user.
- Reliability. - Perform well in any situation.
- Performance: -Provides quick responses for any action.
- Availability: -Available all the time when system is on.
- Portable: -Web Application so portable across any platform.

3.4 HARDWARE REQUIRED FOR THE DEVELOPMENT OF THE PROJECT

- A personal computer
- RAM – 1 GB or more
- Atleast 250 MB hard disk space for installation
- A microphone
- Speakers
- An active Internet Connection

Software required for the development of the project

- PyCharm Community Edition 2017 or any other Python IDE.
- Python 2.7 and above
- A Windows OS – Windows 7, Windows 8, Windows 8.1 or Windows 10.

CHAPTER 4 CONCLUSION AND FUTURE WORK

4.1 CONCLUSIONS

4.1.1 Advantages of Software

- Open different windows software ,based on input voice.
- Useful for the people with disabilities.
- Lower Operational Costs.
- Smart replies to user's queries
- Taps into Google for sports, weather and other information.

4.1.2 Disadvantages of Software

- If the system has to work under noisy environments, background noise may corrupt the original data and leads to misinterpretation.
- If words that are pronounced similar for example, their, there, this technology face difficulty in distinguishing them.

4.2 SCOPE OF FUTURE WORK

This work can be taken into more detail and more work can be done on the project in order to bring additional features to it. The current software does not support large vocabulary, the work can be done in order to bring more words in vocabulary which increases the efficiency of software. Voice recording can be done on this software by storing voice in the database which can also act as reminder. It increases the functionality of a software.

REFERENCES

APPENDIX

1. www.google.com – GOOGLE
2. www.wikipedia.com – WIKIPEDIA
3. <https://www.codeproject.com/articles/483347/speech-recognition-speech-to-text-text-to-speech-a> - CODEPROJECT
4. <https://dev.projectoxford.ai/docs/services/563309b6778daf02acc0a508/operations/5645c3271984551c84ec6797> - VOICE RECOGNITION API