**IFET COLLEGE OF ENGINEERING**

**(AN AUTONOMOUS INSTITUTION)**

**VILLUPURAM**

**MINI PROJECT**

**Subject Code: 19UCSMP301**



**INTELLIGENT DIGITAL ASSISTANT**

By

**MATHUSOOTHANAN .K**

**And**

**DHRUV KRISHNAN .J**

**COMPUTER SCIENCE AND ENGINEERING**

Under the guidance of

**Mrs. Manjubala ASP/CSE**



**IFET COLLEGE OF ENGINEERING**

**(AN AUTONOMOUS INSTITUTION)**

**VILLUPURAM – 605108.**

**BONAFIDE CERTIFICATE**

Certified that this mini project report on **“INTELLIGENT DIGITAL ASSISTANT”** is a bonafide work of **“MATHUSOOTHANAN .K”** who carried out the project work under my supervision.

**SIGNATURE OF THE GUIDE** **HEAD OF THE DEPARTMENT**

The mini project report submitted for the viva voice held on \_\_\_\_\_\_\_\_\_\_

**INTERNAL EXAMINER** **EXTERNAL EXAMINER**

**INTELLIGENT DIGITAL ASSISTANT**

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **CHAPTER NO.** | **TITLE** | **PAGE NO** |
|  | **ABSTRACT** | **5** |
| **1.** | **INTRODUCTION** | **6** |
| **2.** | **SCOPE OF THE PROJECT** | **7** |
| **3.** | **OBJECTIVES** | **8** |
| **4.** | **EXISTING SYSTEM**   * 1. DISADVANTAGES OF EXISTING SYSTEM | **9**  **9** |
| **5.** | **PROPOSED SYSTEM**   * 1. MODULE DESCRIPTION   2. ADVANTAGES OF PROPOSED SYSTEM | **11**  **11**  **15** |
| **6.** | **SYSTEM REQUIREMENTS** | **16** |
| * 1. SOFTWARE REQUIREMENTS | **16** |
| **7.** | **SYSTEM ARCHITECTURE** | **17** |
| **8.** | **RESULTS** | **18** |
| **9.** | **CONCLUSION** | **20** |
|  | **APPENDIX** | **21** |

**ABSTRACT**

In this modern era, day to day life became smarter and interlinked with technology. We already know some voice assistance like google, Siri. etc. Now in our voice assistance system, it can act as a basic medical prescriber, daily schedule reminder, note writer, calculator and a search tool. This project works on voice input and give output through voice and displays the text on the screen. The main agenda of our voice assistance makes people smart and give instant and computed results. The voice assistance takes the voice input through our microphone (Bluetooth and wired microphone) and it converts our voice into computer understandable language gives the required solutions and answers which are asked by the user. This assistance connects with the world wide web to provide results that the user has questioned. Natural Language Processing algorithm helps computer machines to engage in communication using natural human language in many forms.

**CHAPTER 1**

**INTRODUCTION**

Today the development of artificial intelligence (AI) systems that can organize a natural human-machine interaction (through voice, communication, gestures, facial expressions, etc.) are gaining in popularity. One of the most studied and popular was the direction of interaction, based on the understanding of the machine by the machine of the natural human language. It is no longer a human who learns to communicate with a machine, but a machine learns to communicate with a human, exploring his actions, habits, behaviour and trying to become his personalized assistant.

Virtual assistants are software programs that help you ease your day-to-day tasks, such as showing weather reports, creating remainders, making shopping lists etc. They can take commands via text (online chatbots) or by voice. Voice-based intelligent assistants need an invoking word or wake word to activate the listener, followed by the command. We have so many virtual assistants, such as Apple’s Siri, Amazon’s Alexa and Microsoft’s Cortana.

This system is designed to be used efficiently on desktops. Personal assistants’ software improves user productivity by managing routine tasks of the user and by providing information from an online source to the user.

This project was started on the premise that there is a sufficient amount of openly available data and information on the web that can be utilized to build a virtual assistant that has access to making intelligent decisions for routine user activities.

Keywords: Virtual Assistant Using Python, AI, Digital assistance, Virtual Assistance, Python

**CHAPTER 2**

**SCOPE OF THE SYSTEM**

Companies That Hire Virtual Assistants According to the Flex Jobs database, the companies below frequently hire for virtual assistant jobs:

* BELAY
* Boldly
* Equivity
* My BTLR
* Profit Factory
* Randstad
* tawk.to
* VaVa Virtual Assistants

There are many other companies that commonly hire for virtual assistants, too. When it comes to job titles, be on the lookout for job titles like virtual assistant, virtual administrative assistant, administrative coordinator, personal assistant, executive assistant, and similar titles.

**CHAPTER 3**

**OBJECTIVES**

The objective of AI virtual assistants for 2021 is to replace secretaries and personal assistants. Those who perform tasks such as reading text and dictation, finding phone numbers and making calls, emailing schedules and meeting reminders, etc

**CHAPTER 4**

**EXISTING SYSTEM**

Each company developer of the intelligent assistant applies his own specific methods and approaches for development, which in turn affects the final product. One assistant can synthesize speech more qualitatively, another can more accurately and without additional explanations and corrections perform tasks, others can perform a narrower range of tasks, but most accurately and as the user wants. Obviously, there is no universal assistant who would perform all tasks equally well. The set of characteristics that an assistant has depends entirely on which area the developer has paid more attention to.

Since all systems are based on machine learning methods and use for their creation huge amounts of data collected from various sources and then trained on them, an important role is played by the source of this data, be it search systems, various information sources or social networks. The amount of information from different sources determines the nature of the assistant, which can result as a result. Despite the different approaches to learning, different algorithms and techniques, the principle of building such systems remain approximately the same.

|  |  |
| --- | --- |
| **Voice Technology** | **Brain Technology** |
| Voice activation. | Voice bio-metrics. |
| Automatic speech recognition (ASR). | Dialog management. |
| Teach – to – speech  (TTS). | Natural Language understanding (NLU).  Named Entity Recognition (NER). |

the technologies that are used to create intelligent systems of interaction with a human by his natural language. The main technologies are voice activation, automatic speech recognition, Teach-To-Speech, voice biometrics, dialogue manager, natural language understanding and named entity recognition.

**3.1 DISADVANTAGES OF EXISTING SYTEM**

* Leads to lost concentration on the task in hand

Using a virtual assistant with voice recognition can cause you to lose concentration on the tasks in hand. You are giving attention to the virtual assistant when providing instructions and trying to accomplish the task in hand at the same time. When the attention is divided, you are prone to making a mistake. So, before giving instructions to a virtual assistant, make sure to stop whatever you are doing.

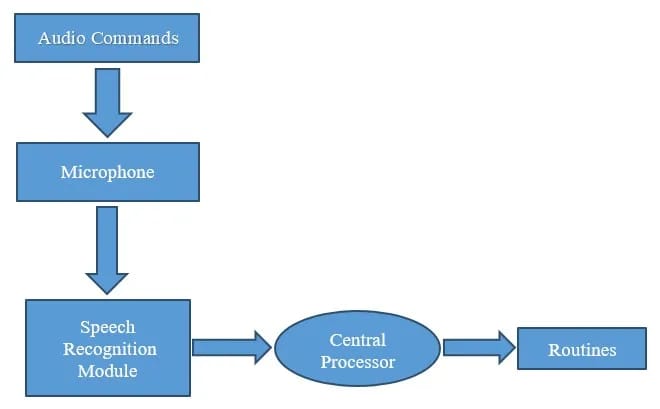
* Privacy Concerns
* Delay timing response

It is dependent on the speed of the internet.

**CHAPTER 4**

**PROPOSED SYSTEM**

The work started with analysing the audio commands given by the user through the microphone. This can be anything like getting any information, operating a computer’s internal files, etc. This is an empirical qualitative study, based on reading above mentioned literature and testing their examples. Tests are made by programming according to books and online resources, with the explicit goal to find best practices and a more advanced understanding of Voice Assistant.



The above pic shows the workflow of the basic process of the voice assistant. Speech recognition is used to convert the speech input to text. This text is then fed to the central processor which determines the nature of the command and calls the relevant script for execution.

But the complexities don’t stop there. Even with hundreds of hours of input, other factors can play a huge role in whether or not the software can understand you. Background noise can easily throw a speech recognition device off track. This is because it does not inherently have the ability to distinguish the ambient sounds it “hears” of a dog barking or a helicopter flying overhead, from your voice. Engineers have to program that ability into the device; they conduct data collection of these ambient sounds and “tell” the device to filter them out. Another factor is the way humans naturally shift the pitch of their voice to accommodate for noisy environments; speech recognition systems can be sensitive to these pitch changes.

**5.1 MODULE DESCRIPTION**

**Speech Recognition module:**

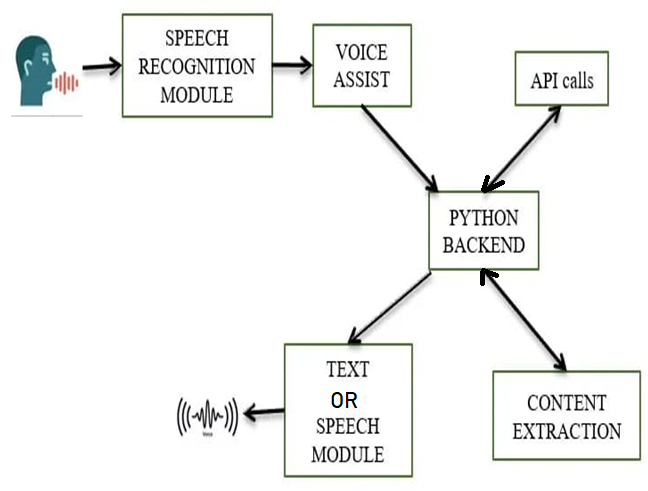
The system uses Google’s online speech recognition system for converting speech input to text. The speech input Users can obtain texts from the special corpora organized on the computer network server at the information centre from the microphone is temporarily stored in the system which is then sent to Google cloud for speech recognition. The equivalent text is then received and fed to the central processor.

**Python Backend:**

The python backend gets the output from the speech recognition module and then identifies whether the command or the speech output is an API Call and Context Extraction. The output is then sent back to the python backend to give the required output to the user.

**API calls:**

API stands for Application Programming Interface. An API is a software intermediary that allows two applications to talk to each other. In other words, an API is a messenger that delivers your request to the provider that you’re requesting it from and then delivers the response back to you.



**Content Extraction:**

Context extraction (CE) is the task of automatically extracting structured information from unstructured and/or semi-structured machine-readable documents. In most cases, this activity concerns processing human language texts using natural language processing (NLP). Recent activities in multimedia document processing like automatic annotation and content extraction out of images/audio/video could be seen as context extraction TEST RESULTS.

**Text-to-speech module:**

Text-to-Speech (TTS) refers to the ability of computers to read text aloud. A TTS Engine converts written text to a phonemic representation, then converts the phonemic representation to waveforms that can be output as sound. TTS engines with different languages, dialects and specialized vocabularies are available through third-party publishers.

**5.2 ADVANTAGES OF PROPOSED SYSTEM**

* **Saves Time**

Assigning repeated tasks to virtual voice assistants frees up the human time and resources. They also perform these mundane tasks more efficiently than the humans, which often lead to increased customer satisfaction. With these virtual voice assistants dealing with routine tasks, the human agents can actually spend their time on other calls where human intervention is required. The agents can also utilize that time on selling and thus adding value to BFSI. Organisations normally roll out only a fraction of their customer base to these virtual assistants and once they start to learn and grow the database, more workload is shifted.

* **Hands-Free Operation**

These digital voice-enabled assistants also make the operations hands-free. You don’t need to touch any screen or type in anything, which virtually saves time. Users love to use these assistants for a variety of reasons and efficiency is one of the top reason. People find them to be much more efficient than when such tasks are entrusted to humans.

* **Ease of Use**

The next spot was taken by the ease of use. 50% of those who were surveyed said that it is a better interaction medium and they find it easier than any other method of communication used. There were also 40% people who said that they can utilise that time for completing some other tasks simultaneously.

Consumers love to make voice payments as they find the transactions to be happening faster and easier. It also adds a layer of security as voice recognition is used rather than typing in passwords. Data suggests that voice payments are seeing a Compound Annual Growth Rate (CAGR) of 31% in the US alone. It looks like the voice payment revolution has started and businesses are adopting it because the consumers are trying it in higher-than-expected numbers.

**CHAPTER 6**

**SYSTEM REQUIREMENTS**

Recommended Operating Systems

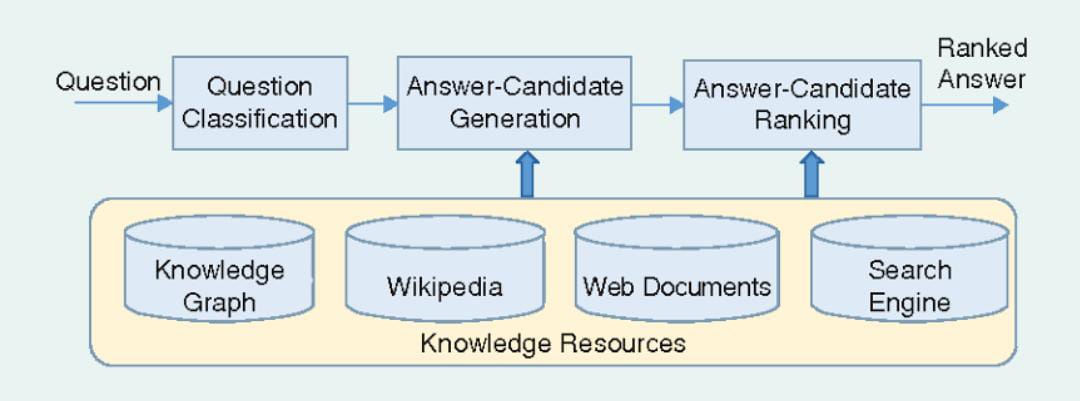
* **Windows:**7 or newer
* **MAC:** OS X v10.7 or higher
* **Linux:** Ubuntu

We strongly recommend a computer fewer than 5 years old.

* **Processor**: Minimum 1 GHz; Recommended 2GHz or more
* Ethernet connection (LAN) OR a wireless adapter (Wi-Fi)
* **Hard Drive:** Minimum 32 GB; Recommended 64 GB or more
* **Memory (RAM)**: Minimum 1 GB; Recommended 4 GB or above

**CHAPTER 8**

**SYSTEM ARCHITECTURE**:



**CHAPTER 9**

**RESULT**

**CHAPTER 10**

**Conclusion**

In this paper “Virtual Assistant Using Python” we discussed the design and implementation of Digital Assistance. The project is built using open-source software modules with PyCharm community backing which can accommodate any updates shortly. The modular nature of this project makes it more flexible and easier to add additional features without disturbing current system functionalities.

It not only works on human commands but also give responses to the user based on the query being asked or the words spoken by the user such as opening tasks and operations. It is greeting the user the way the user feels more comfortable and feels free to interact with the voice assistant. The application should also eliminate any kind of unnecessary manual work required in the user life of performing every task. The entire system works on the verbal input rather than the next one.

**APPENDIX**

import speech\_recognition as sr

import pyttsx3

import datetime

import wikipedia

import webbrowser

import os

import time

import subprocess

import pywhatkit

import pyjokes

print('Loading your personal assistant - IDA')

engine=pyttsx3.init('sapi5')

voices=engine.getProperty('voices')

engine.setProperty('voice',voices[1].id)

def speak(text):

    engine.say(text)

    engine.runAndWait()

def wishMe():

    hour = datetime.datetime.now().hour

    if hour>=0 and hour<12:

        speak("Hello,Good Morning")

        print("Hello,Good Morning")

    elif hour>=12 and hour<18:

        speak("Hello,Good Afternoon")

        print("Hello,Good Afternoon")

    else:

        speak("Hello,Good Evening")

        print("Hello,Good Evening")

def takeCommand():

    r = sr.Recognizer()

    with sr.Microphone() as source:

        print("Listening...")

        audio = r.listen(source)

        try:

            statement = r.recognize\_google(audio,language='en-in')

            print(f"user said:{statement}\n")

        except Exception as e:

            speak("Pardon me, please say that again")

            return "None"

        return statement

speak("Loading your personal assistant IDA")

wishMe()

if \_\_name\_\_ == '\_\_main\_\_':

    while True:

        speak("Tell me how can I help you now?")

        statement = takeCommand().lower()

        if statement == 0:

            continue

        if "good bye" in statement or "ok bye" in statement or "stop" in statement:

            speak('your personal assistant IDA is shutting down,Good bye')

            print('your personal assistant IDA is shutting down,Good bye')

            break

        if 'wikipedia' in statement:

            speak('Searching Wikipedia...')

            statement = statement.replace("wikipedia", "")

            results = wikipedia.summary(statement, sentences=3)

            speak("According to Wikipedia")

            print(results)

            speak(results)

        elif 'open youtube' in statement:

            webbrowser.open\_new\_tab("https://www.youtube.com")

            speak("youtube is open now")

            time.sleep(5)

        elif 'open google' in statement:

            webbrowser.open\_new\_tab("https://www.google.com")

            speak("Google chrome is open now")

            time.sleep(5)

        elif 'open gmail' in statement:

            webbrowser.open\_new\_tab("gmail.com")

            speak("Google Mail open now")

            time.sleep(5)

        elif 'open notepad' in statement:

            name = statement.replace('open', '')

            path0 = f"%windir%\\system32\\notepad.exe"

            os.system(path0)

        elif 'open wordpad' in statement:

            name = statement.replace('open', '')

            path0 = f"%windir%\\system32\\wordpad.exe"

            os.system(path0)

        elif 'open word' in statement:

            name = statement.replace('open', '')

            path0 = f"C:\\Program Files\\Microsoft Office\\root\\Office16\\WINWORD.EXE"

            os.system(path0)

        elif 'open powerpoint' in statement:

            name = statement.replace('open', '')

            path0 = f"C:\\Program Files\\Microsoft Office\\root\\Office16\\POWERPNT.EXE"

            os.system(path0)

        elif 'time' in statement:

            strTime=datetime.datetime.now().strftime("%H:%M:%S")

            speak(f"the time is {strTime}")

        elif 'who are you' in statement or 'what can you do' in statement:

            speak('I am IDA your personal assistant. I am programmed to do minor tasks like'

                  'opening youtube,google chrome,gmail and predict time,')

        elif "who made you" in statement or "who created you" in statement or "who discovered you" in statement:

            speak("I was built by DK AND MADHU")

            print("I was built by DK AND MADHU")

        elif 'news' in statement:

            news = webbrowser.open\_new\_tab("https://timesofindia.indiatimes.com/home/headlines")

            speak('Here are some headlines from the Times of India,Happy reading')

            time.sleep(6)

        elif 'play' in statement:

            song = statement.replace('play', '')

            speak('playing ' + song)

            pywhatkit.playonyt(song)

        elif 'joke' in statement:

            speak(pyjokes.get\_joke())

        elif 'search' in statement:

            statement = statement.replace("search", "")

            webbrowser.open\_new\_tab(statement)

            time.sleep(5)

        elif "log off" in statement or "sign out" in statement:

            speak("Ok , your pc will log off in 10 sec make sure you exit from all applications")

            subprocess.call(["shutdown", "/l"])

        elif"weather" in statement:

             speak("finding the weather")

             webbrowser.open\_new\_tab("https://www.windy.com/?12.900,80.221,5")

time.sleep(3)