e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:05/Issue:12/December-2023 Impact Factor- 7.868 www.irjmets.com

SMART VIRTUAL VOICE ASSISTANT USING PYTHON

Prof. Rashmi Kannake[*1], Pranmya Kale[*2], Nikhil Dongre[*3], Vipul Kshirsagar[*4], Yogesh Tajane[*5]

[*1]Asst. Professor, Department Of Computer Science And Engineering, Wainganga College Of Engineering And Management, Nagpur, Maharashtra, India.

[*2,3,4,5]B.Tech Student, Department Of Computer Science And Engineering, Wainganga College Of Engineering And Management, Nagpur, Maharashtra, India.

DOI : https://www.doi.org/10.56726/IRJMETS47227

ABSTRACT

One popular topic these days is the voice assistant, which is essentially a program that is intended to listen intently to human voice commands and then respond with comprehensible text. This allows people to communicate with computers or other electronic devices. Voice assistants are a ubiquitous feature of modern life, providing great assistance when we are running around with ever-more-busy schedules. In the contemporary landscape, an overwhelming proportion of global denizens have readily integrated Voice Assistants into their daily lives. This widespread adoption is evident from ubiquitous manifestations such as Google's smartphone-based voice assistant, a user-friendly interface that even young children, aged as little as five, can proficiently navigate, largely attributable to the exigencies imposed by the recent global pandemic, which has significantly intensified reliance on smartphones. Moreover, Voice Assistants such as Amazon's Alexa have emerged as indispensable tools, capable of facilitating diverse tasks ranging from user entertainment to the operation and management of household appliances, effectively engaging with the burgeoning realm of the Internet of Things (IoT). Notably, one of the paramount attributes of these Voice Assistants is their profound inclusivity, serving as invaluable aids to individuals with physical disabilities. To elucidate, those who encounter mobility challenges can harness the IoT functionality to control and oversee household devices, thereby enhancing their autonomy and overall quality of life. Hence, there exists a compelling impulsion to cultivate and advance the development of Voice Assistants, to perpetuate their utility for an increasingly discerning and diverse user base. This although comes with a credible yet abstruse difficulty as the establishment of connectivity to the user base in terms of non-linearity in network connections or absence of data linkup facilities. Thus, this project aims to manifest a tool that can, in many terms, be able to secure reliable results for the completion of issues.

Keywords: Voice Assistant, Python, Virtual Assistant, Jarvis.

I. INTRODUCTION

Innovations determine our lives on digital platforms. One of the most important inventions for improving our lives and enabling hands-free use is an intelligent personal assistant. In our daily lives, speech recognition technology is tremendously helpful in a variety of settings and applications. Our goal is to create a PC personal assistant that can respond to voice commands and carry out user inquiries. The development of machine learning and artificial intelligence has led to the development of intelligent personal assistants, which help to automate and simplify daily tasks. One such personal assistant is "JARVIS," a sophisticated artificial intelligence system that is capable of carrying out a wide range of duties, including managing smart home appliances and appointment scheduling. JARVIS is not restricted to a single user or sector of the economy. Rather, it is a flexible and adaptive instrument that can be tailored to individual and corporate needs. JARVIS can comprehend and react to voice commands thanks to machine learning and natural language processing technology. The smooth integration of JARVIS with other programs and services is one of its main benefits.

1.1 Overview

There are already several virtual assistants available for mobile devices. With advancing technology, we are now able to introduce more sophisticated solutions into our daily lives. Technology continues to progress, and people are increasingly reliant on it, especially when it comes to using computers. We all want to enhance the usability of our computers, and while the traditional way to communicate with a computer is through a keyboard, a more convenient method is to issue commands through voice. This process involves recognizing

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:05/Issue:12/December-2023 Impact Factor- 7.868 www.irjmets.com

spoken words and responding with a synthetic speech in real time. These virtual assistants enable users to automate a variety of tasks, including but not limited to email management, task organization, and controlling media playback. They understand and respond to natural language voice commands, making tasks easier for users. These virtual assistants rely on technologies such as machine learning, natural language processing, and speech recognition platforms. They use complex algorithms to learn from the input they receive and continuously improve their ability to anticipate and meet the needs of the end user. These desktop assistants act as intelligent mediators between users and their computers, with the primary objective of simplifying and streamlining tasks, which in turn saves time and enhances overall efficiency. Furthermore, these assistants harness the power of artificial intelligence (AI) and natural language processing (NLP) techniques to comprehend natural language queries, making interactions more intuitive and conversational.

1.2 Speech Recognition

Speech recognition, often called speech-to-text, is when a machine or program can listen to spoken words and turn them into written text that we can read. But this kind of technology has its limits. It can only understand words and phrases when they're spoken clearly and not too complicated. However, after going through these four steps, a software program transforms the sound captured by a microphone into written language that both computers and humans can comprehend: 1. Analyse the audio. 2. Divide it into segments. 3. Convert it into a format that computers can read. 4. Utilize an algorithm to find the most appropriate written representation

1.3 Virtual Assistant

A virtual assistant is a freelance professional who offers administrative support to clients, all while working from a location separate from the client's office. Typically, a virtual assistant operates from their home, although they may have access to essential tools like shared calendars from anywhere. Common tasks performed by virtual assistants include scheduling appointments, making phone calls, arranging travel plans, and handling email accounts.

II. METHODOLOGY

2.1 Methodology and Workflow

The project aims to design and implement Jarvis, a personal voice assistant using Python, to provide users with voice-activated services, similar to popular voice assistants like Siri and Alexa. The assistant will be able to perform the required tasks following a procedure as explained further. It will be developed in several phases including a proper web or desktop implementation for the same:

1. Requirements Gathering and Design: Define the scope and functionality of Jarvis. Identify the key features and capabilities, such as voice recognition, natural language processing, and task execution. Design the user interface and user experience for voice interactions.
2. Speech Recognition: Implement a speech recognition module to convert audio input into text. Choose or build a suitable speech recognition API or library.
3. Natural Language Processing (NLP): Develop an NLP engine to understand and interpret user queries. Implement intent recognition to determine user intentions. Build a knowledge base or integrate with external APIs for information retrieval.
4. Voice Synthesis: Create a text-to-speech (TTS) module for generating voice responses. Select a TTS library or service for voice output.
5. Task Execution: Implement functionalities for executing tasks based on user requests. Develop plugins or modules for common tasks like setting alarms, sending messages, or providing weather updates.
6. User Interface: Design a user-friendly interface for interaction with Jarvis. Implement voice-triggered commands and responses. Create a graphical interface (optional).
7. Integration: Integrate Jervis with third-party services and APIs for expanded functionality. Ensure compatibility with various platforms and devices.
8. Deployment: Deploy Jervis on appropriate platforms (e.g., desktop, mobile, or embedded devices). Provide ongoing maintenance and updates.
9. Future Enhancements: Consider future features and upgrades to improve the usability and use cases of the product for improved conversation handling. Explore partnerships with other services to expand

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:05/Issue:12/December-2023 Impact Factor- 7.868 www.irjmets.com

capabilities. Developing a personal voice assistant is a complex and ongoing project that requires a mix of skills in natural language processing, machine learning, software development, and user experience design.

10. Constant learning and adaptation: Make use of machine learning techniques to enable your assistant to develop and adapt over time. For instance, it can be trained to recognize voice patterns, user preferences, and frequently used commands to provide a personalized experience.

2.2 Command Flow Diagram

The following figure expresses the working flow chart of the assistant program that is to be implemented in the project flow. The flow starts with the invoking of the voice assistant by the runtime program. The program opens up with greeting the user in the process of speaking through the speaker array. It then accepts the user input command, processes it through the interpreter, and next derives the conclusion to accept or reject the assignment. It calls the required functions thereby running the required program and thus ending the task after a successful completion. The workflow returns for the next command to be invoked by the user.

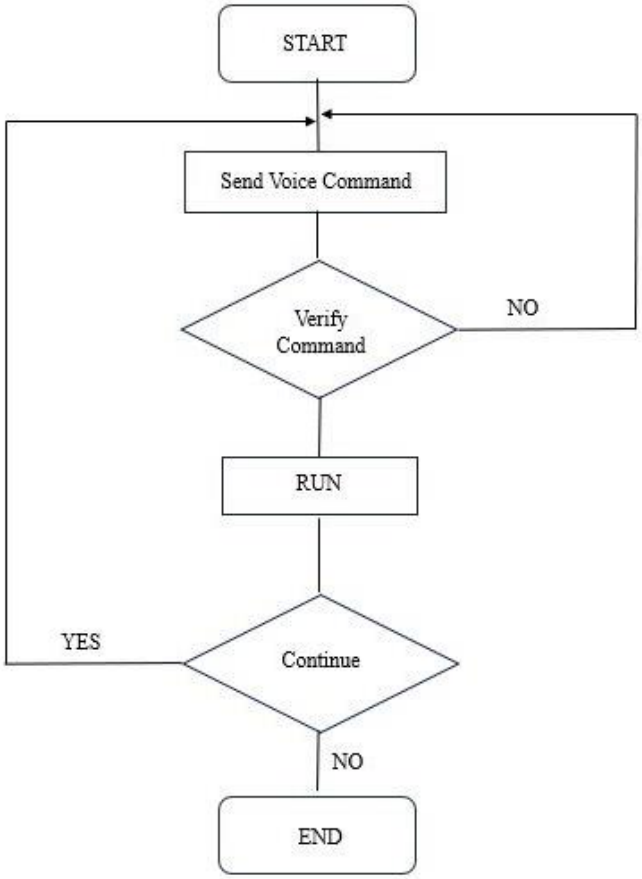

Fig. 2.1 Flow Diagram

III. SYSTEM ARCHITECTURE

3.1 System Diagram

The complete system architecture can be described into a series of steps required for the processing of the entire user command. Thus, the architecture includes the internal processing structure of the model.

1. Speech Recognition: The system listens to what you say and turns it into written words. When you talk into the microphone, the system first saves what you say on a computer. Then, it sends that saved speech to Google's computer system to figure out what words you spoke. After Google figures it out, the system gets those words back and gives them to the main computer to use.
2. Python Backend: The Python backend analyzes the voice recognition module's results to identify if it's an API call, context extraction, or system call. It then returns this information to the Python backend, which processes it to deliver the requested outcomes to the user.

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:05/Issue:12/December-2023 Impact Factor- 7.868 www.irjmets.com

3. API Calls: An API stands for Application Programming Interface, which is a software tool enabling two applications to talk to each other. Think of it as a messenger; it forwards your request to the provider and brings back the response.
4. Content Extraction: Context extraction (CE) involves converting unstructured or semi-structured machine-readable content into structured data. This typically involves using natural language processing (NLP) on human language texts. Recent applications of CE include multimedia document tasks like automatically adding descriptions and extracting content from images, audio, and video, as evident in test results.
5. System Calls: A system call is when computer software asks the operating system's core (kernel) for services. It can involve hardware tasks like accessing a hard drive, creating new processes, or interacting with kernel functions like process scheduling. Essentially, system calls act as the software's bridge to communicate with the operating system.
6. Text-to-Speech: The ability of computers to speak written text is known as text-to-speech (TTS). It involves converting written text into a phonetic form, which is then transformed into sound waveforms by a TTS Engine. These TTS engines, available from third-party providers, support multiple languages, dialects, and specialized vocabularies.

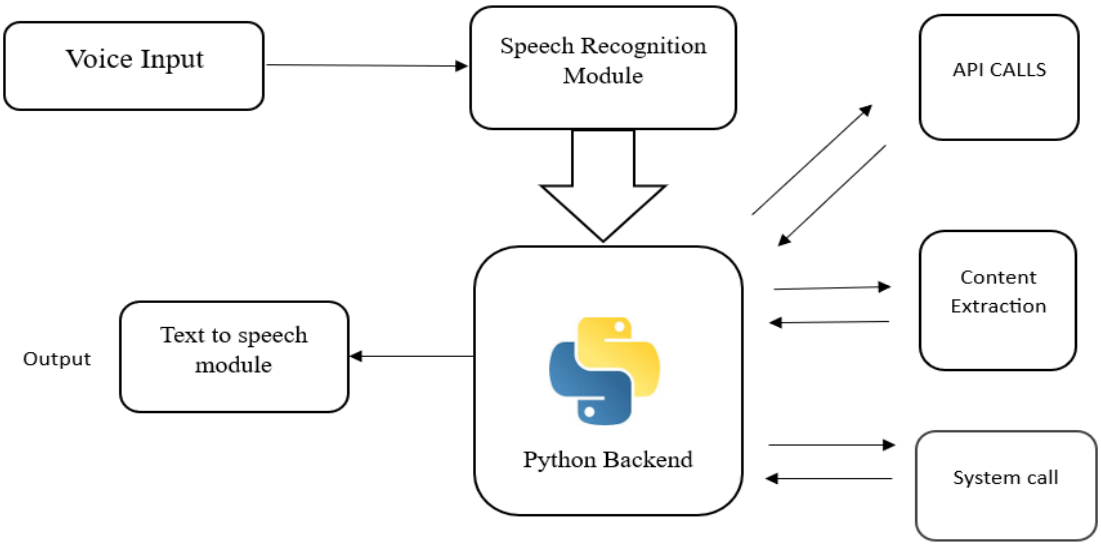

Fig. 3.1 System Architecture

3.2 Sequence Diagram

The Sequence Diagram for the project is as follows. The user first inputs the voice command, which when recognized by the microphone is sent to the interpreter. Then the web scrapper scrapes the web for the relevant answers and solutions. It passes the solution to the speaker who subsequently speaks out the results in the form of voice due to the text-to-speech library.

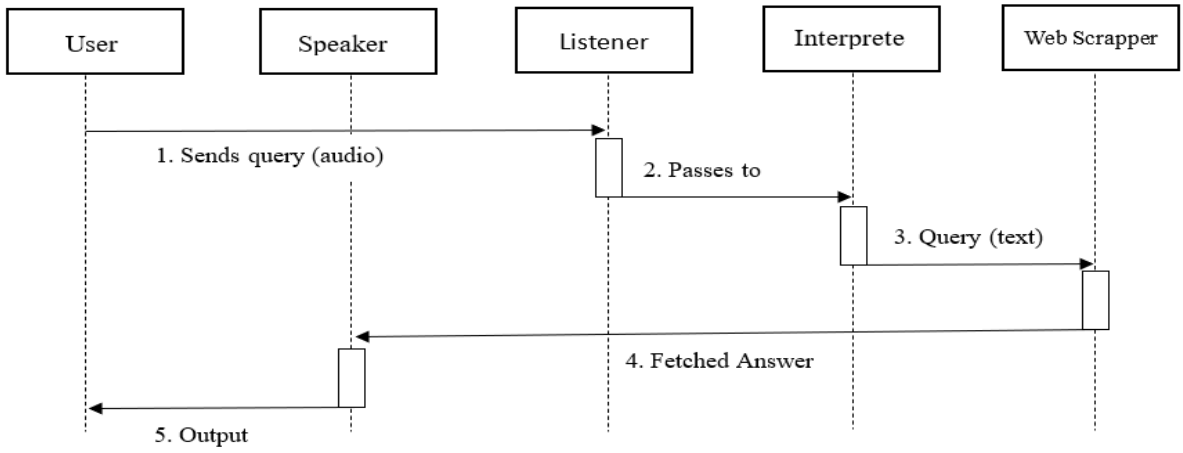

Fig. 3.2 Sequence Diagram

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:05/Issue:12/December-2023 Impact Factor- 7.868 www.irjmets.com

IV. RESULTS AND DISCUSSION

4.1 Results

The output shows a conversation with the model for procuring proper results as expected. The outputs are based on informal conversations, remembering texts, opening web searches, etc.

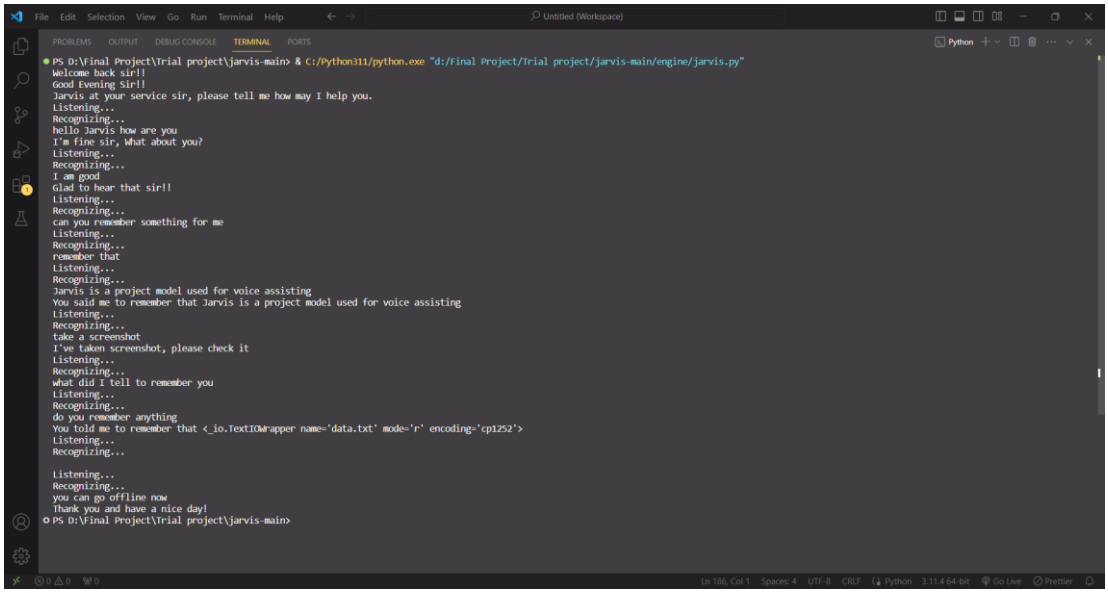

Fig 4.1

4.2 Testing Results

1. Weather report test

Performed a weather test to verify the weather report feature. The objective of the test case is to verify that the virtual voice assistant can fetch and deliver the weather report.

The test case includes a command to the assistant and the expected output:

1. Speak "What's the weather like in CITY" or "What's the weather like today".
2. The virtual voice assistant should deliver an accurate weather report.

The expected output is that the virtual voice assistant delivers accurate weather. The assistant passes the test.

2. Email sending test

Performed email test to verify email sending feature.

The test case includes a command to the assistant:

1. Speak "send email".
2. Provide the recipient's email address, subject, and body of the email.
3. The virtual voice assistant should send the email and reply with a success or error message.

The expected output is that the virtual voice assistant sends the email successfully. The test case is passed.

3. Opening websites test

Performed website opening test to verify that it can open websites.

The test case includes a command to the assistant:

1. Speak "open (website name)".
2. The virtual voice assistant should open websites.

The expected output is that the virtual voice assistant opens the website. The test case is passed.

V. CONCLUSION

In conclusion, the voice assistant developed in this project is capable of performing various tasks such as browsing the internet, sending emails, generating images, and interacting with the user through conversation. It can do so by utilizing various APIs and technologies such as stability_sdk, Google Speech Recognition, and SMTP. The voice assistant is also able to perform system tasks such as opening and closing tabs, windows, and

applications, as well as taking screenshots and manipulating text in the clipboard. There are several potential areas for future enhancement for the voice assistant. One possibility is to improve the natural language processing capabilities of the chatbot model, to enable more seamless conversation with the user enable more seamless conversation with the user.

VI. REFERENCES

[1] K. Noda, H. Arie, Y. Suga, T. Ogata, Multimodal integration learning of robot behavior using deep neural networks, Elsevier: Robotics and Autonomous Systems, 2014.

[2] Nivedita Singh, Dr. Diwakar Yagyasen, Mr. Surya Vikram Singh, Gaurav Kumar, and Harshit Agrawal, this idea is employed utilizing Python, Machine Learning, and AI, in IJIRT, 2021.

[3] Deepak Shende, Ria Umahiya, Monika Raghorte, Aishwarya Bhisikar, Anup Bhange, "AI-Based Voice Assistant Using Python", Journal of Emerging Technologies and Innovative Research (JETIR), February 2019, Volume 6, Issue 2.

[4] Dilawar Shah, Tuul Triyason, Debajyoti Pal, and Vajirasak Vanijja, on Usability of Voice-based Intelligent Personal Assistants, in IEEE Xplore in May 2021.

[5] Rajdeep Paul and Nirmalya Mukhopadhyay, on A Novel Python-based Voice Assistance System for Reducing the Hardware Dependency, in International Research Journal of Engineering and Technology (IRJET), in May 2021.

[6] John Levis and Ruslan Suvorov, "Automatic Speech Recognition".

[7] V. Geetha, C.K. Gomathy, Manasa Sri Vardhan, and Pavan Kumar, on The Voice-Enabled Personal Assistant for PC using Python, at International Journal of Engineering and Advanced Technology (IJEAT), in April 2021.

[8] Rahul Kumar, Garima Saruprіa, Varshil Panwala, Smit Shah, and Nehal Shah, on Power-Efficient Smart Home with Voice Assistant on International Journal of Engineering and Advanced Technology (IJEAT), in April 2020.

[9] Ravikumar N R, Pratik C, Satvik Bhandar, Rahul Kumar, Mayura D Tapkire on Virtual Voice Assistant in International Research Journal of Engineering and Technology in April 2020.

[10] Dimitrios Buhalis and Moldavska, on Voice Assistant in hospitality: using Artificial Intelligence for Customer Service, in Journal of Hospitality and Tourism Technology in April 2022.

[11] Subhash S, Prajwal N Srivatsa, S Siddhesh, A Ulhas, and B Santosh, in IEEE Xplore as a conference in Artificial Intelligence Based Voice Assistant in July 2020.