***Virtual assistant Jarvis using python***

Anish Bochare1, Arya Deshpande2, Shruti Ghate3, Ansh Ghodke4, Yogesh Gaikwad5.

Department of Information Technology,

MAEER’s MIT Polytechnic , Pune.

anishbochare@gmail.com1 aryadeshpande3004@gmail.com 2,

shrutighate7@gmail.com 3, anshghodke29@gmail.com 4, yogesh.gaikwad@mitwpu.edu.in5

**Abstract—**

**The Jarvis Python project aims to develop a virtual assistant capable of various tasks through voice commands. The assistant will use speech recognition and natural language processing to understand user requests and respond with appropriate actions. The project will involve integrating various APIs, such as weather and news, to provide real-time information to the user. The assistant will also be capable of setting reminders, managing to-do lists. The ultimate goal of the Jarvis Python project is to create a reliable and efficient virtual assistant that can simplify daily tasks and enhance the user experience.**

**The project will involve integrating third-party APIs and services to give the assistant the ability to carry out a variety of activities, including playing music, setting alarms, and managing smart home gadgets. The assistant will also be customizable, allowing users to add new features and functionality based on their specific needs. The ultimate goal of the Jarvis Python project is to create a powerful and versatile virtual assistant that can improve productivity and enhance the user experience.**

1. **INTRODUCTION**

Jarvis is a fictional character in the Marvel Cinematic Universe, who is the artificial intelligence system created by Tony Stark, also known as Iron Man. Jarvis serves as a loyal assistant and friend to Tony Stark, providing him with information, managing his technology and helping him in various tasks. Jarvis is an acronym for "Just A Rather Very Intelligent System", Although Jarvis is a fictional character, the concept of artificial intelligence systems like him is becoming increasingly prevalent in our daily lives. Siri, Alexa, and Google Assistant are examples of virtual assistants, that can assist with tasks and provide information. JARVIS, standing for "Just A Really Very Intelligent System," is an acronym. Jarvis is a popular virtual assistant built using Python programming language. It enables users to utilise voice commands to communicate with their computer. and perform various tasks such as opening applications, browsing the web, sending emails, and more.

There are several libraries available in Python that can be used to build Jarvis you need to have a basic understanding of Python and its libraries. Once you have set up the required libraries, you can start building the assistant by defining functions to perform specific actions and integrating them with the speech recognition and text-to-speech conversion modules.

Overall, Jarvis is a fun project that allows you to explore the capabilities of AI and voice recognition technology.

1. **OVERVIEW OF AI**

Jarvis is a popular open-source virtual assistant program written in Python that uses AI to provide various services, such as voice-activated commands, scheduling tasks, and more.

In the context of Jarvis, AI is used to enable natural language processing (NLP) and speech recognition capabilities, which allows users to communicate with the program using spoken or written commands. The program uses machine learning algorithms to continuously improve its ability to understand and respond to user inputs, making it more accurate and efficient over time.



1. **LITERATURE SURVEY**

first literature paper check is “Artificial Intelligence- grounded Voice Assistant” by Subhash, Prajwal N Srivatsa, S Siddesh, A Ullas, B Santhosh. We've added a lot to our design in comparison to other backing. The fact that it's a hands-free operation makes it relatively useful in diurnal life. It's a remarkably simple operation. It is also used in business settings, similar as laboratories where workers must class while wearing body suits and gloves for protection. Yet, they may pierce any information by using a voice adjunct, which makes their job easier.

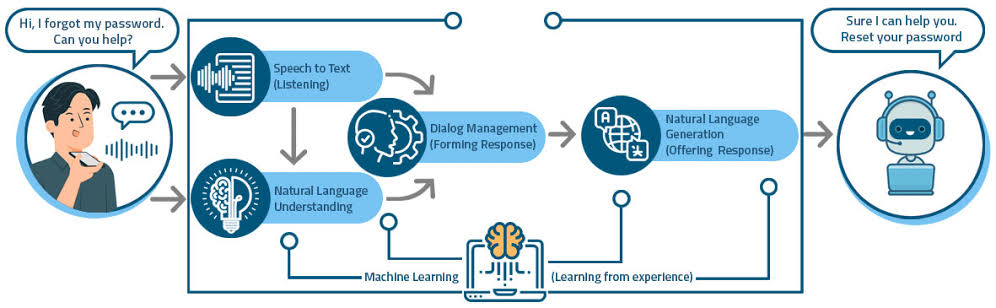
Alternate paper for literature check is “Short exploration on Voice Control System Grounded on Artificial Intelligence Assistant” by Tae- Kook Kim. The design of a voice control system using bedded bias and open API AIs is discussed in this paper. The proposed system is a voice command system that activates modules with a stoner's spoken commands. Drug addicts can create their own system using the helpful speech recognition interface and various modules provided by this system.

Third paper for literature check is “Next- generation of virtual particular sidekicks Microsoft Cortana, Apple Siri, Amazon Alexa and Google Home ” by Veton Këpuska, Gamal Bohouta. This offer introduces the structure of the Coming- Generation of Virtual particular sidekicks, which is a new VPAs system with a coherent structure designed to discourse with a mortal. In both the input and affair channels, this VPAs system communicated using speech, plates, videotape, etc. By utilizing tools like image/videotape recognition, speech recognition, and the Knowledge Base, the VPAs system will also be used to enhance stoner-computer commerce.

1. **ARCHITECTURE**

Jarvis is a well-known virtual assistant who can carry out a variety of tasks, including providing answers and setting reminders. The architecture of Jarvis can vary depending on how it is implemented, but a common architecture might involve the following components:

Wake word detection: Jarvis needs to be able to listen for a wake word (such as "Hey Jarvis" or "OK Google") that signals that the user wants to interact with the virtual assistant. This component typically involves audio processing and machine learning algorithms to recognize the wake word.



**System Architecture**

Initially, the condition here is whether or not Jarvis voice assistant is active; if it is active, it requests user input; otherwise, make Jarvis active (make it on). The user then provides input in the form of speech or text; if the input is in text, it proceeds to the action to be taken or the skill to be executed; if the input is in speech, it uses the speech recognition feature to convert it to text and proceeds to the action.

Now, if the skills to be executed are adequate to Jarvis, it responds to the user in the form of speech and then executes the commands for operations.

1. ***Speech recognition***

After the wake word is detected, Jarvis needs to be able to recognize the user's speech and convert it into text. This component also typically involves audio processing and machine learning algorithms, such as deep neural networks, to perform the speech recognition.

1. ***Natural language understanding***

Once the user's speech is converted into text, Jarvis needs to be able to understand the user's intent and extract relevant information from the user's request. This part typically uses sentiment analysis and named entity recognition in natural language processing.

1. ***Dialog management***

After understanding the user's intent, Jarvis needs to be able to manage a dialog with the user to gather any additional information needed to fulfill the user's request. This component typically involves a rule-based or machine learning-based approach to dialog management.

1. ***Task fulfillment***

Once Jarvis has gathered all the necessary information from the user, it needs to be able to perform the requested task. This could involve calling APIs for external services, controlling smart home devices, or simply providing information to the user.

1. ***Response generation***

Finally, Jarvis needs to be able to generate a response to the user, either as speech or as text. This component typically involves natural language generation techniques such as text-to-speech synthesis or template-based text generation.

Overall, the architecture of Jarvis is a complex system that requires expertise in several areas, including audio processing, natural language processing, and software engineering

* **PYTHON LIBRARIES**

In JARVIS following python libraries were used:

pyttsx3: to convert text to speech this library is used.

Speech Recognition: to convert speech to text this library is used.

pywhatkit: To launch WhatsApp communication with required features.

Datetime: System date and time is used with the help of this library.

Smtplib: Between mail servers, emails are routed using the SMTP protocol.

pyPDF2: This module is used for reading, writing of pdf files.

Pyjokes: It is a Python library with so many entertaining jokes in it..

Webbrowser: It offers drug users an interface for displaying web-based documents.

Pyautogui: Python libraries are being used to generate the graphical stoner interface.

1. **FLOWCHAT**

Diagram

Description automatically generated

1. **RESULT**







1. **FUTURE SCENARIO**

In the future, it is likely that Jarvis will continue to evolve and become even more sophisticated. Here are a few potential scenarios for the future of Jarvis Python:

Personalized Recommendations: With access to large amounts of data about a user's preferences, Jarvis Python could become better at making personalized recommendations. For example, based on a user's viewing preferences, Jarvis Python could suggest motion pictures or television shows. or suggest recipes based on their dietary preferences.

Integration with Business Applications: As more businesses adopt AI and automation technologies, Jarvis Python could be integrated with various business applications to help automate tasks such as scheduling meetings, responding to emails, or even managing inventory.

Enhanced Natural Language Processing: With advances in natural language processing (NLP) and machine learning, Jarvis Python could become even better at understanding and responding to natural language queries. This could make it easier for users to communicate with Jarvis Python and get the information they need more quickly and accurately.

Overall, the possibilities for Jarvis Python are virtually limitless. As technology continues to advance, it will be exciting to see how this virtual assistant evolves and adapts to meet the needs of its use

1. **CHALLENGES**

Despite the potential for growth and development, Jarvis Python also faces several challenges. Here are a few examples:

Privacy Concerns: Jarvis Python will likely need to collect a significant amount of personal data from users in order to function effectively. This could raise privacy concerns among some users, particularly if their personal information is compromised in a security breach.

Language and Cultural Barriers: Jarvis Python will need to be able to understand and respond to queries in multiple languages and dialects in order to be useful on a global scale. Additionally, cultural differences mayrequire adaptations in order to ensure that the technology is culturally appropriate and sensitive.

Ethical Considerations: As with any AI-based technology, there are ethical considerations surrounding the use of Jarvis Python. For example, how will the technology be used in hiring decisions or in determining creditworthiness? These issues will need to be addressed in order to ensure that Jarvis Python is used fairly and responsibly.

1. **FEATURES OF JARVIS**

Clap Detection to Start: Clap detection is a popular method of starting devices such as lights or music players by simply clapping. To implement this feature, you would need to use a microphone to detect the sound of a clap, and then use a program or script to trigger an action based on that sound. You could use a library such as PyAudio in Python to capture the sound, and then use a simple algorithm to detect the sound of a clap. Once the clap is detected, you can use another script or program to perform the desired action, such as turning on a light or playing music.

Jarvis Detection: Jarvis is a popular AI assistant that is used to automate tasks and provide assistance to users. To create a Jarvis detection system, you would need to use a voice recognition library such as SpeechRecognition in Python to capture and analyze the user's speech. You could then use a Natural Language Processing (NLP) library such as NLTK or spaCy to understand the user's intent and determine what action should be taken. Once the user's intent is understood, you can use another script or program to perform the desired action, such as setting a reminder or sending an email.

Speed Test: A speed test is a useful tool for measuring the speed of an internet connection. To create a speed test, you would need to use a network testing library such as speedtest-cli in Python to measure the download and upload speeds of the internet connection. You could then use a simple algorithm to calculate the average speed, and display the results to the user.

Notepad Creation: To create and edit text files, use the text editor Notepad. To create a notepad in Python, you could use the built-in tkinter library to create a graphical user interface (GUI) that allows the user to enter and edit text. You could then use another script or program to save the text entered by the user to a file.

Phone Control for Jarvis Python: To create a phone control system for Jarvis, you would need to use a library such as pybluez in Python to communicate with the phone via Bluetooth. You could then use a simple algorithm to detect the user's voice commands and perform the desired action on the phone, such as making a call or sending a text message. You could also use a GUI library such as tkinter to create a graphical user interface that allows the user to interact with the phone using buttons and other controls.

1. **CONCLUSION**

The project is a great example of how to build a virtual assistant using Python. By integrating various APIs and libraries, it is possible to create a powerful tool that can perform a variety of tasks.

In this project, we have seen how to use the speech recognition and text-to-speech conversion libraries to enable the assistant to listen to commands and respond with voice output. We have also explored how to use APIs for various services such as weather, news, and web search.

Furthermore, we have discussed implementing basic natural language processing techniques to understand user input and respond appropriately. We have also shown how to use conditional statements and loops to create a robust decision-making process for the assistant.

Overall, the Jarvis Python project is a great starting point for anyone interested in building a virtual assistant. It provides a solid foundation for further customization and expansion and can be adapted to meet a wide range of needs and use cases.

1. **REFERENCES**

1.N. H. Tandel, H. B. Prajapati and V. K. Dabhi, "Voice Recognition and Voice Comparison using Machine Learning Techniques: A Survey," 2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, India, 2020, pp. 459-465, doi: 10.1109/ICACCS48705.2020.9074184.

2.R. Sangpal, T. Gawand, S. Vaykar and N. Madhavi, "JARVIS: An interpretation of AIML with integration of gTTS and Python," 2019 2nd International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT), Kannur, India,2019,pp.486-489,doi:10.1109/ICICICT46008.2019.8993344.

3.V. Këpuska and G. Bohouta, "Next-generation of virtual personal assistants (Microsoft Cortana, Apple Siri, Amazon Alexa and Google Home)," 2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, USA, 2018, pp. 99-103, doi: 10.1109/CCWC.2018.8301638.

4.K. N., R. V., S. S. S. and D. R., "Intelligent Personal Assistant - Implementing Voice Commands enabling Speech Recognition," 2020 International Conference on System, Computation, Automation and Networking (ICSCAN), Pondicherry, India, 2020, pp. 1-5, doi: 10.1109/ICSCAN49426.2020.9262279.

5.S. Subhash, P. N. Srivatsa, S. Siddesh, A. Ullas and B. Santhosh, "Artificial Intelligence-based Voice Assistant," 2020 Fourth World Conference on Smart Trends in Systems, Security and Sustainability (WorldS4), London, UK, 2020, pp. 593-596, doi: 10.1109/WorldS450073.2020.9210344.

6.Harshit Agrawal, Nivedita Singh, Gaurav Kumar, Dr. Diwakar Yagyasen, Mr. Surya Vikram Singh. "Voice Assistant Using Python" An International Open Access-revied, Refereed Journal.Unique Paper ID: 152099, Publication Volume & Issue: Volume 8, Issue 2, Page(s): 419-423

7.Deepak Shende. Ria Umabiya, Monika Raghorte, Aishwarya Bhisikar. Anup Bhange. "Al Based Voice Assistant Using Python", International Journal of Emerging Technologies and Innovative Research (www.jetir.org), ISSN 2349-5162, Vol.6, Issue 2, page no.506-509, February-2019.

8.[4]. Tulshan, Amrita & Dhage, Sudhir. (2019). “Survey on Virtual Assistant: Google Assistant, Siri, Cortana, Alexa”, 4th International Symposium SIRS 2018, Bangalore, India, September 19–22, 2018, Revised Selected Papers. 10.1007/978-981-13-5758-9\_17.[5]. Dr. Ksh

9 T. -K. Kim, "Short Research on Voice Control System Based on Artificial Intelligence Assistant," 2020 International Conference on Electronics, Information, and Communication (ICEIC), Barcelona, Spain, 2020, pp. 1-2, doi: 10.1109/ICEIC49074.2020.9051160.

10. Y. J. Gaikwad, N. Potnis, S. Jadhav and Y. Ahire, "File Sharing Application with OTP Service," 2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA), Coimbatore, India, 2018, pp. 1464-1467, doi: 10.1109/ICECA.2018.8474858.