



INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

AI Based Voice Assistant

Siddharth Singh, Devansh Singh, Dr. Arvind Jagtap

Student, Student, Professor
Computer Science and Engineering,
MIT School of Computing, Pune, India

Abstract: This paper describes the design and implementation of a Python voice assistant that can perform multiple user-directed tasks like opening websites, sending WhatsApp messages, and fetching information from Wikipedia. The assistant is powered by speech recognition, natural language processing, and text-to-speech synthesis to communicate with users in real-time. The system is implemented with Python libraries like speech_recognition, pyttsx3 and pywhatkit combined with logic to respond to user queries dynamically. Experimental use demonstrates that the assistant can consistently perform predefined tasks, laying the groundwork for more sophisticated conversational AI applications.

Index Terms - Voice Assistant, Python, Speech Recognition, Natural Language Processing, Automation

I. INTRODUCTION

Voice assistants have become a ubiquitous feature of contemporary human-computer interaction, transforming the manner in which users interact with digital devices. Well-known systems like Amazon Alexa, Google Assistant, and Apple Siri have shown the promise of voice-controlled automation in increasing user convenience and accessibility. These systems take advantage of advances in speech recognition, natural language processing (NLP), and artificial intelligence to understand spoken instructions and execute a range of tasks from information retrieval to device control.

This work introduces the construction of a light voice assistant implemented with Python to execute simple but practical functions like the opening of YouTube, sending WhatsApp messages, and extracting Wikipedia summaries. Differing from typical commercial voice assistants, the project here emphasizes simplicity, customization, and educational potential and is hence applicable to academic settings and personal productivity applications.

The assistant is built on top of Python libraries such as `speech\_recognition`, `pyttsx3`, `pywhatkit`, and `wikipedia`. The system records user voice, interprets the speech to determine intent, and carries out the appropriate actions. With modular structure and simplicity of integration, the assistant illustrates how natural language interfaces can be achieved through open-source tools with minimal infrastructure needed.

The rest of this paper is structured as follows: Section II discusses related work and technologies. Section III outlines the system architecture. Section IV explains the implementation. Section V gives the results and evaluation. Section VI concludes the paper with future possibilities for enhancement.

Ease of Use

II.OBJECTIVE

The main goal of this project is to create and implement a Python voice assistant that can execute simple automated tasks through natural language voice commands. These tasks involve opening websites like YouTube, sending messages on WhatsApp, and fetching summaries from Wikipedia. The assistant should show how open-source Python libraries can be efficiently used to develop a simple, customizable, and interactive voice-controlled system that increases user accessibility and productivity.

III.TOOLS AND LANGUAGES

The building of the voice assistant was made up of numerous tools and programming libraries, which were predominantly located within the Python environment. The below tools and technologies were used:

Programming Language:

Python 3.x – Selected because it is easy, readable, and supports a huge range of speech and automation libraries.

Python Libraries and Frameworks:

speech_recognition – For speech-to-text conversion with the help of online and offline recognition engines.

pyttsx3 – For text-to-speech, supporting the assistant to reply to users through speech.

pywhatkit – To execute functions like sending a WhatsApp message or opening YouTube.

wikipedia – For fetching concise information as the answer to a user query.

datetime – For performing date and time-related functions.

webbrowser – For launching URLs in the default browser.

os – For accessing and running operating system-related tasks.

Development Environment:

Visual Studio Code / PyCharm – Used as the integrated development environment (IDE) for writing and debugging the code.

Operating System:

Windows 10/11 – The project was created and tested using a Windows platform

IV. LITERATURE REVIEW

There are studies that indicate the use of voice-based systems in automation. Current commercial voice assistants are highly dependent on cloud infrastructure, whereas the present project aims for a light local system. Past research justifies the usability of open-source libraries for speech-to-text and text-to-speech operations in Python, opening the avenue for creating customized assistants with low dependencies.

V. METHODOLOGY

5.1: Technologies Used

Python 3.x: Programming language.

Libraries:

speech_recognition: Voice to text conversion.

pyttsx3: Text-to-speech functionality.

pywhatkit: Sends WhatsApp messages, plays YouTube videos.

wikipedia: Retrieves summaries of topics.

os: Executes system-level commands.

5.2: System Workflow

Microphone captures the user's voice.

Speech is converted to text using Google's API.

Assistant parses command and triggers the respective function.

System responds using pyttsx3 or carries out task.

5.3: Functional Modules

WhatsApp Messaging:

Uses `pywhatkit.sendwhatmsg_instantly()` for instant messaging.

Wikipedia Search:

Retrieves summaries using `wikipedia.summary(topic)`.

System Operations:

Executes `shutdown/restart` using `os.system("shutdown /s /t 1")` etc.

Web Browsing:

Open YouTube, Google using `webbrowser.open()`.

VI. RESULTS

The voice assistant was also tried in a Windows 10 environment. It performed the following:

Identified and executed more than 90% of everyday English commands.

Sent real-time WhatsApp messages successfully.

Effectively returned Wikipedia summaries.

Performed system operations correctly.

Open websites using the default browser.

VII. CONCLUSION

The voice assistant that was created is a hands-on illustration of speech recognition being utilized to communicate with software applications. With limited hardware and open-source software, a working assistant can be made to enhance user productivity and convenience. Enhancements in the future could be through AI-based learning for contextual comprehension and integration with IoT devices.

VIII. FUTURE WORK

Integration with cloud APIs for support of more languages.

Including email and calendar management functionality.

Improving security for WhatsApp and system commands.

Deployment on mobile or Raspberry Pi for portability.

IX. ACKNOWLEDGMENT

The author feels it is important to convey sincere appreciation to the departmental teaching staff of Computer Science and Engineering at MIT School of Computing for their regular encouragement and support during the course of developing this project. Special appreciation goes to Prof. Arvind Jagtap for constructive inputs and suggestions. The project was undertaken as a part of the undergraduate syllabus and had no external funding.

X. REFERENCES

- [1] [1] G. Hinton, L. Deng, D. Yu, et al., “Deep Neural Networks for Acoustic Modeling in Speech Recognition,” *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, Nov. 2012.
- [2] [2] D. Jurafsky and J. H. Martin, *Speech and Language Processing*, 3rd ed., Pearson, 2023. [Online]. Available: <https://web.stanford.edu/~jurafsky/slp3/>
- [3] [3] A. M. Turing, “Computing Machinery and Intelligence,” *Mind*, vol. 59, no. 236, pp. 433–460, 1950.
- [4] [4] Python Software Foundation, “SpeechRecognition 3.10.0,” [Online]. Available: <https://pypi.org/project/SpeechRecognition/>
- [5] [5] K. Tanwar, “Building a Voice Assistant in Python,” *Medium*, 2021. [Online]. Available: <https://medium.com/@kirtanwar/building-a-voice-assistant-in-python-3b36e5b54aaf>
- [6] [6] “pyttsx3 2.90 Documentation,” [Online]. Available: <https://pyttsx3.readthedocs.io/>
- [7] [7] “pywhatkit Documentation,” [Online]. Available: <https://pypi.org/project/pywhatkit/>
- [8] [8] Wikipedia contributors, “Wikipedia,” *Wikipedia, The Free Encyclopedia*, [Online]. Available: <https://en.wikipedia.org/>