International Journal of Engineering Technology and Management Sciences
Website: ijetms.in Issue: 3 Volume No.8May - June – 2024
DOI:10.46647/ijetms.2024.v08i03.014 ISSN: 2581-4621

PROTO – A Customized Assistant To Optimize Personal Tasks

T. Raghavendra Gupta [1], N. Naga Tanusri[1], R. Rishikesh Reddy[3], V. Sandeep Kumar [4], Neha Basvoju[5]

[,1]*Associate Professor, Computer Science & Engineering, Hyderabad Institute of Technology and Management,India.*

[2]*Student, Computer Science & Engineering, Hyderabad Institute of Technology and Management, India.*

[3]Student, *Computer Science & Engineering, Hyderabad Institute of Technology and Management, India.*

[4]Student, *Computer Science & Engineering, Hyderabad Institute of Technology and Management, India.*

[5]Student, *Computer Science & Engineering, Hyderabad Institute of Technology and Management, India.*

***Abstract*—** In today's interconnected world, the combination of technology and AI has become an essential part of daily life. AI-powered voice assistants have evolved into indispensable tools, seamlessly integrating technology into everyday tasks. This project demonstrates the development of a Customised Artificial Intelligence Assistant in Python, which was chosen for its extensive library set suitable for building AI applications. This assistant is designed to make computer-related tasks easier and more convenient for users. The assistant improves functionality and user experience by combining voice recognition, natural language processing, and external API integration, such as the OpenAI API. What distinguishes this assistant is its high level of customisation and extensibility, which allows users to tailor its behaviour and features to their specific requirements. These intelligent programmes use voice commands to understand human natural languages and complete tasks for the user. It can perform a variety of basic tasks, including launching applications, conducting searches without a web browser, playing music, and more, all by responding to voice commands. Furthermore, the assistant works seamlessly with popular tools and services, providing a user-friendly experience for both beginners and experts. The goal of this project is to improve users' productivity and efficiency in their daily computer tasks by combining accessibility, automation, and customisable features.

Keywords—Automation Tools, Customizable AI Assistant, Natural Language Processing, OpenAI API Integration, User Productivity, Voice Recognition.

INTRODUCTION

In today's digitalized world, voice searches have surpassed text searches, with mobile device web searches recently outpacing those conducted on computers. Analysts predict that by 2025, 50% of searches will be conducted by voice. Virtual assistants are becoming more intelligent, able to handle emails, detect intent, extract critical information,
automate processes and provide personalised responses. This technology is extremely popular due to its ability to perform a variety of tasks, including checking the date and time, searching the web, opening specific applications, and sending greetings.

Virtual assistants boost productivity by allowing users to control computers with voice commands, saving time and increasing functionality. These as reminders and useful tools for a variety of tasks, such as displaying weather reports, setting reminders, and creating shopping lists. Voice recognition technology, also known as Automatic Speech Recognition (ASR), converts

DOI:10.46647/ijetms.2024.v08i03.014 ISSN: 2581-4621

spoken words into computer-readable formats by accepting and processing user input via voice and providing feedback in the form of actions or search results. Intelligent Personal Assistants (IPAs) such as Amazon Alexa, Microsoft Cortana, Google Assistant, and Apple Siri enable hands-free interaction, allowing users to do things like search for information, schedule meetings, and make phone calls. Natural Language User Interfaces (NLUI) enable H2M and HCI by translating human intentions into device control commands via speech recognition. This hands-free interaction opens new possibilities for computing applications in health, sports, education, and entertainment.

Imagine a surgeon in the middle of an operation seamlessly accessing a patient's medical history via a simple voice command, or an athlete receiving real-time feedback on performance metrics without interrupting their workout. In education, IPAs can transform learning environments by delivering interactive, voice-guided lessons that adapt to students' pace and comprehension. Even in entertainment, the impact is profound—AI-powered platforms such as Spotify curate personalised music experiences that learn and evolve in response to the user's preferences. As voice interaction becomes the primary mode of human-computer interaction, the potential applications for IPAs grow, improving efficiency and natural communication in everyday tasks. This evolution is about more than just convenience; it's about creating a more intuitive, responsive, and personalised digital ecosystem that closely resembles how humans naturally communicate and interact.

LITERATURE SURVEY

Moustafa Elshafei et al. [1] posits that Virtual Personal Assistants (VPAs) are the future of mobile and smart user network services. VPAs facilitate access to a broad spectrum of information upon user requests, simplifying the management of tasks and appointments and enabling control of phone calls via voice commands. A significant feature of VPAs is the task manager, accessible through voice interaction or login, which helps users enhance time management and minimize distractions.

Subash. S et al. [2] developed an AI-based virtual assistant suitable for both desktops and mobile devices. This assistant translates spoken content into readable data and then converts the necessary information into speech using the pyttsx3 module.

Ankit Pandey et al. [3] created a smart voice assistant capable of making notes, exchanging emails, and scheduling calendar meetings. Designed for user convenience, this assistant allows monitoring of appliances via speech commands and gathers necessary information efficiently.

Jianliang Meng et al. [4] and colleagues offered an overview of speech recognition technology, focusing on vocal input as the research subject. Their study examines how machines automatically recognize and interpret user voice inputs through recognition pattern modules and speech signal processing.

Emad S. Othman et al. [5] presented a personal voice assistant using a microcontroller like Raspberry Pi, managing various tasks for user convenience. The assistant includes substructure configuration and performs effectively with minimal space-time complexity.

METHODOLOGY

PROTO's system architecture is intended to allow users to interact with their desktop systems in a smooth and intuitive manner, utilising speech recognition, text interpretation, API integration, and text-to-speech capabilities. The Speech Recognition Module is at the heart of PROTO, capturing audio input from the user via a microphone. This module translates spoken language to text, which serves as the system's initial input.

The Voice Assistant Module interprets the text produced by the Speech Recognition Module. It compares interpreted commands to predefined commands stored in the system and decides the

DOI:10.46647/ijetms.2024.v08i03.014 ISSN: 2581-4621

appropriate action based on the user's input. This module serves as the decision-making engine for PROTO, allowing it to respond intelligently to user questions and requests. All paragraphs must be indented. All paragraphs must be justified, i.e. both left-justified and right-justified. It includes the API Calls Module, which improves communication with external APIs. These APIs are used to retrieve information or execute certain operations, such as retrieving the most recent weather prediction, gathering data from web sources, or controlling external devices. This integration improves PROTO's ability to give consumers with real-time, contextually relevant information.

Once a response or action is determined, the Text to Speech Module turns the system's text output into natural-sounding speech, making the response audible and allowing for smooth contact with the user. This module provides clear and comprehensible voice output. The Python Backend serves as the foundation for PROTO's development by providing the infrastructure required to integrate and manage the many modules. It ensures that the various components operate and interact smoothly, hence sustaining the stability and reliability of the system.

The system architecture also includes the Content Extraction Module, which is not directly connected to the other modules in the design. This module is responsible for extracting relevant content from external sources in response to user inquiries, hence improving the system's ability to give complete and reliable information.

A. Tools and Libraries

- *pyttsx3*: Text-to-speech conversion library for Python, supporting multiple platforms and voices.

- *SpeechRecognition:* Enables Python programs to recognize speech from audio input using various engines and APIs.

- *datetime:* Provides classes to manipulate dates and times in Python applications.

- *os:* Allows Python programs to interact with the operating system, facilitating file and system operations.

- *webbrowser:* High-level interface to display web-based documents, opening URLs in new browser windows or tabs.

- *requests:* Simplifies sending HTTP requests in Python, handling headers, form data, and response data.

- *time:* Provides time-related functions for code execution delay and time measurement.

- *OpenAI:* Access OpenAI's GPT models and tools through their REST API for advanced AI capabilities.

- *subprocess:* Creates and interacts with subprocesses, executing external commands and retrieving their output.

- *ctypes:* Interacts with shared libraries in C-compatible data types, useful for integrating with C-based libraries.

- *json:* Supports JSON data interchange format, parsing JSON strings and encoding Python objects.

International Journal of Engineering Technology and Management Sciences
Website: ijetms.in Issue: 3 Volume No.8May - June – 2024
DOI:10.46647/ijetms.2024.v08i03.014 ISSN: 2581-4621

- *psutil:* Cross-platform library for retrieving system utilization and process information (CPU, memory, disks, network).

- *Tkinter:* Built-in Python library for creating GUI applications with widgets and tools for desktop interfaces.

Incorporating these tools and libraries into PROTO's architecture ensures flexibility, scalability, and reliability, making it a powerful tool for enhancing user productivity and interaction with desktop systems through voice commands.

IMPLEMENTATION

PROTO is a powerful desktop voice assistant designed to improve user interactions and productivity by utilising natural language processing and automation. PROTO, built in Python with the PyCharm IDE, combines numerous critical components to provide seamless voice-controlled functionality.

It starts up when users say a wake-up word, such as "Hey PROTO!", which causes the voice assistant to start listening for commands using a connected microphone. The assistant uses Speech to Text (STT) technology to translate spoken words into text, which is then processed and analysed to determine the user's intent.

When the user's command is recognised, the PROTO backend processes it using a Python-based architecture. Commands are analysed to identify specific actions, which are then efficiently interpreted and executed using natural language processing algorithms.

For example, PROTO can launch applications such as Chrome or the calculator, receive real-time information such as weather updates and news, and execute a variety of additional tasks in response to user requests. It also works with external APIs like NewsAPI and OpenWeatherMap to return relevant data in response to user queries. This technology enables PROTO to give real-time information on news headlines, weather forecasts, and other dynamic material using voice queries.

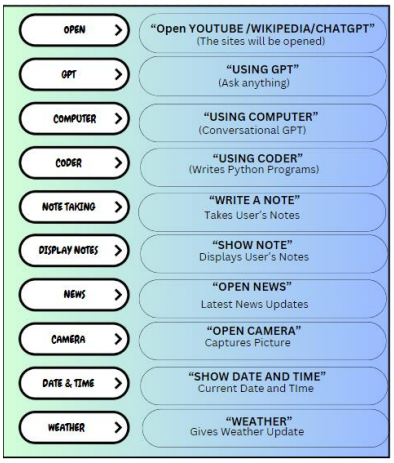

Figure 1: PROTO's Command List

In addition to information retrieval, PROTO provides system control via OS and Web Browser modules. Users can manage their desktop environment by using voice commands to restart the machine or open programmes. The user interface is intended to encourage intuitive engagement, with PROTO activating upon hearing a wake-up phrase such as "Hey PROTO!" Users interact with a linked microphone, which converts speech to text using Speech to Text (STT) technology. This

International Journal of Engineering Technology and Management Sciences
Website: ijetms.in Issue: 3 Volume No.8May - June – 2024
DOI:10.46647/ijetms.2024.v08i03.014 ISSN: 2581-4621

text is processed by the backend, which uses natural language processing algorithms to efficiently interpret requests and carry out activities. PROTO's comprehensive error handling provides reliability by handling unexpected inputs and errors, hence improving overall speed and user experience.

The application of PROTO demonstrates its versatility and efficiency as a desktop voice assistant, utilising powerful speech recognition, natural language processing, and integration with external APIs to improve user productivity and information accessibility. Future enhancements aim to broaden PROTO's capabilities, keeping it at the cutting edge of voice assistant technology.

RESULTS & DISCUSSIONS

In this section, we present the results and discussion focusing on the functionalities of two commands- using GPT and CODER. Analysing these examples, we aim to provide insights into the effectiveness and versatility of command-based functionalities in achieving PROTO's objectives and meeting user requirements.

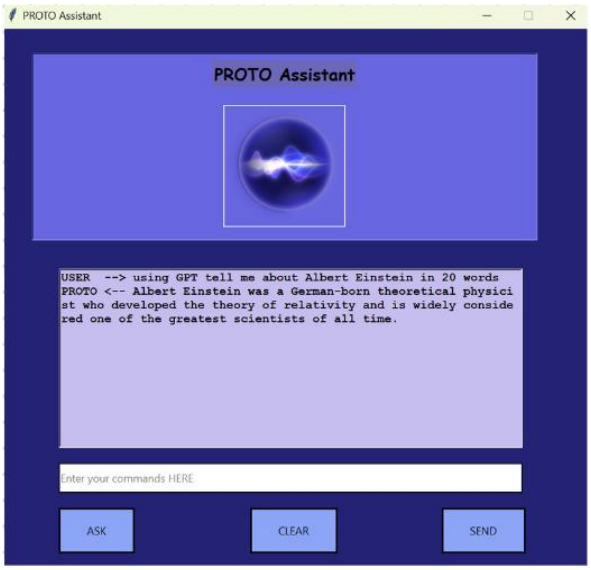

Figure 2: Working of PROTO using GPT voice command

Figure 2 demonstrates that with the GPTcommand, users have the capability to inquire about diverse topics and request information, showcasing the command's versatility in responding to various queries.

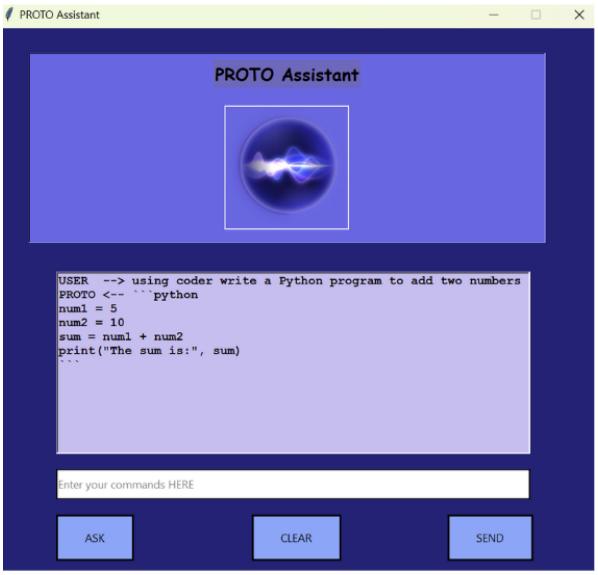

Figure 3: Working of PROTO using CODER voice command.

DOI:10.46647/ijetms.2024.v08i03.014 ISSN: 2581-4621

Figure 3 shows users to define outcomes, provide code and solve any problem in Python using coder command.

CONCLUSION

PROTO is a substantial development in desktop voice assistants, providing a combination of natural language processing and programming skills. As technology advances, PROTO's functionality and uses will be refined and expanded, making it an invaluable tool for users looking to interact with technology in a more intuitive and efficient manner. However, changes are required to solve issues like accuracy in complex queries and syntax problems in programming jobs. Future development could focus on strengthening PROTO's natural language processing skills, including more complex AI models, and boosting the coder's accuracy and reliability. Additionally, increasing the list of supported programming languages and adding debugging features would increase PROTO's usability.

References

[1] Moustafa Elshafei. Virtual personal assistant (vpa) for mobile users. Mitel Networks (2000–2002), 2002.
[2] S Subhash, Prajwal N Srivatsa, S Siddesh, A Ullas, and B Santhosh. Artificial intelligence-based voice assistant. In 2020 Fourth world conference on smart trends in systems, security and sustainability (WorldS4), pages 593–596. IEEE, 2020.
[3] Ankit Pandey, Vaibhav Vashist, Prateek Tiwari, Sunil Sikka, and Priyanka Makkar. Smart voice based virtual personal assistants with artificial intelligence. Artificial Computational Research Society, 1(3), 2020.
[4] Jianliang Meng, Junwei Zhang, and Haoquan Zhao. Overview of the speech recognition technology. In 2012 fourth international conference on computational and information sciences, pages 199–202. IEEE, 2012.
[5] Emad S Othman et al. Voice controlled personal assistant using raspberry pi. Inter- national Journal of Scientific & Engineering Research, 8(11):1611–1615, 2017.