

Virtual Assistance using Python

Pranav Ambhore¹, Pavan Ingole², Ritesh Shinde³, Vedant Jadhav⁴, Prof. S. L. Dawkhari⁵

Students, Department of Information Technology¹

Professor, Department of Information Technology^{2,3,4,5}

Sinhgad College of Engineering, Pune, Maharashtra, India

Abstract: The paper explores the potential of utilizing new technology to develop an intelligent Virtual Assistant that is capable of using natural language processing and user-based data. It examines existing intelligent programs with different categories of support and evaluates the potential usefulness of a particular software as a Virtual Assistant. The proposed Virtual Assistant should be able to communicate socially through natural language processing, store and analyze user data, and operate without the need for human input or programming. The paper suggests that with advancements in technology, creating virtual personal assistants could become a reality. The authors conducted experiments on a specific software and performed user testing, which demonstrated that a basic program with natural language processing algorithms could already be viable. Overall, the paper presents the idea of an intelligent Virtual Assistant that could revolutionize the way we interact with technology.

Keywords: Python programming language, Natural Language Processing (NLP), Wolfram Alpha API, desktop assistants, machine learning, text-to-speech, speech-to-text, language processing, voice recognition, artificial intelligence, Internet of Things (IoT), and virtual assistants.

I. INTRODUCTION

The primary objective of this project is to create a straightforward and self-contained application that can assist individuals with limited technical expertise or computer literacy in using their computers. As computers have become increasingly affordable and indispensable in daily life, this application aims to provide a solution for those who may feel overwhelmed by technology.

The application functions in a manner similar to popular virtual assistants such as Siri and Google Assistant, with a focus on providing assistance for personal computers. Users can access the virtual assistant through voice commands, keyboard inputs, or remotely via the internet.

The system comes with a set of predefined commands, and users have the option of adding their own custom commands. The virtual assistant can notify users of incoming emails, provide information about the weather and their location, and even reveal IP addresses, MAC addresses, and wireless network passwords. Additionally, users can use voice commands to perform system tasks like shutting down or putting their computer to sleep.

The application also includes a note-taking feature. The system boasts an interface that can take inputs and produce outputs. For various tasks, such as time management and search, there are separate modules. The interface utilizes these modules to execute commands and perform tasks in an efficient and user-friendly manner. The ultimate goal is to make computer usage simpler for individuals with limited technical expertise and promote ease of use.

This project centers around the development of a Windows-based personal assistant utilizing voice recognition technology. The application provides a range of services and functions, including mail sending, event handling, location services, music player support, weather checks, Google and Wikipedia search engines, camera integration, and Bluetooth headset compatibility. By integrating several commonly used desktop services, this program has the potential to significantly enhance daily life and make it more convenient. Moreover, it can be especially helpful for individuals with physical disabilities who may face challenges with manual operations. This program was inspired by Apple's popular virtual assistant, "Siri," which is part of the reason why it was chosen as the subject for a diploma project.

This application was initially launched in conjunction with the release of the iPhone4S. It's a highly engaging and user-friendly tool with broad applicability and considerable potential for further development. It's not limited to any particular generation or profession, and can be deployed across a wide range of industries. However, as it's currently

only available for Apple's iOS, we've introduced a "Virtual Assistant" for personal computers to address this limitation. Voice assistance is particularly valuable for personal assistants, navigation, driving, and supporting individuals with disabilities, among other use cases.

Wikipedia describes "Siri" as an intelligent personal assistant and knowledge navigator designed for use with Apple's iOS. The application leverages a natural language user interface to answer questions, offer recommendations, and perform actions by delegating requests to a set of web services. According to Apple, the software adapts to individual user preferences over time, personalizing results and completing tasks such as identifying nearby restaurants and providing directions.

II. LITERATURE SURVEY

Speech recognition technology has evolved through several waves of major innovations and has now become a standard feature on smartphones and wearable devices, enabling dictation, search, and voice commands. The goal of designing a compact, efficient, and accurate large vocabulary speech recognition system for computers has been a major focus of research. A set of techniques have been developed to improve the performance of automated voice search services, especially for Windows users accessing these services through portable devices.

Machine learning and natural language processing techniques have been refined, and applications that provide speech-based services and content have emerged. As computers become smaller and more ubiquitous, such as with wearables and Internet of Things (IoT) devices, speech recognition technology is becoming more important. The primary function of a speech recognizer is to convert spoken words into text or other forms of data.

One example of a widely used speech recognition application is "SIRI" on iPhones. SIRI serves as a personal assistant with voice recognition intelligence, enabling users to communicate with their mobile devices using voice commands. SIRI processes the user's input and provides responses in various forms, such as performing an action or providing information. This proposed system can change the way users interact with their devices and make it more convenient for Windows users.

Cortana, a desktop-based virtual assistant, offers a range of features including reading messages, tracking location, monitoring browsing history, checking contact lists, and integrating this data to suggest helpful information if allowed. Alternatively, users can type their queries or requests if they prefer not to speak out loud.

Siri, which has been part of iOS since its launch in 2011 with iOS 5, initially provided basic features such as weather and messaging. However, it has expanded significantly to support more third-party integrations with MacOS. While Siri is known for its humor, the virtual assistant is becoming more capable with time. It can now make calls, send messages, schedule meetings, launch apps and games, play music, answer questions, set reminders, and provide weather forecasts. Google Assistant is different from Cortana and Siri. It has incorporated the features of the older Google Now, which is being phased out.

A. Background of the Industry Based

The basis of this project stems from Apple's popular application, "Siri," which was launched alongside the iPhone4S. This user-friendly application has a wide range of real-world applications and great potential for development, making it suitable for many different industries. However, the application is not available for Windows, prompting the development of a "Virtual Assistant" for personal computers. The voice-assisted technology has proven useful for personal assistants, navigational tools, driving aids, and the disabled community, among others. Siri is an intelligent personal assistant that serves as an iOS application for Apple devices. It uses a natural language interface to answer questions, provide recommendations, and perform tasks by accessing web services. Apple claims that the software learns and adapts to the user's preferences over time, resulting in personalized results such as restaurant recommendations and directions.

B. Information Retrieval

The program encompasses a range of functions and services, including mail exchange, mathematical calculations, location services, music player service, weather checking, Google and Wikipedia search engines, camera, Bluetooth

headset support, IP address finding, Saved Wi-Fi passwords finding, and a help menu. Each function has specific information and requirements, as outlined below.

Mail exchange allows customers to send emails to individuals in their contacts. A correct command containing the mail request keyword and destination person should result in the recipient receiving the email.

Location services enable users to check their current location or find a desired destination. Depending on the request category, the user should receive an easily understandable map with the relevant locations.

The music player service offers users the ability to play a specific or randomly selected song from the prestored song list on their desktop PC. The music can be stopped at any time.

The weather checking function allows users to check the weather in any location, with information such as temperature and humidity included. Additionally, users can check the weather for the current day, tomorrow, or the next four days.

The program includes a Google search engine that allows users to search for anything on Google, with the results displayed on their web browser. There is also a Wikipedia search engine that enables users to search for anything on Wikipedia, with the searched content displayed on the browser.

The camera function calls the mobile phone camera to take a picture of the current view, which is then stored in the gallery for later viewing and operation.

To overcome the issue of voice recognition being difficult when the music player is playing or the surroundings are noisy, the program offers Bluetooth headset support, allowing users to speak to the headset rather than the PC if enabled.

The program also features an application opening and closing service, enabling users to open any installed applications such as MS Word, MS PowerPoint, Paint, Notepad, IDEs, Text Editors, File Explorer, and more.

For programmers and IT personnel, the program includes an IP address service that allows users to find the hostname and IP address of their PC, as well as a MAC address finding service.

A wireless network password finding service is also available to find saved WiFi passwords, while a note writing service uses the speech-to-text module for easy Dictation.

The program also includes a joke service for entertainment purposes, a time and date service for telling the current time, date, or day, and a calculation service for mathematical operations such as multiplication, division, sin, cos values, and more.

C. Theory Model

This project encompasses various aspects of software engineering principles, software development models, Python programming skills, APIs, and network communication technologies. The use of APIs and web services in this project is focused on the Wolfram Alpha API, eliminating the need for developers to write additional code as the API handles the execution. Therefore, API serves as an important concept and guides the development theory

The Wolfram Alpha Web service API provides a web-based API that facilitates the integration of Wolfram Alpha's computational and presentation capabilities into web, mobile, desktop, and enterprise applications. Wolfram Alpha is an API that utilizes Wolfram's algorithms, knowledgebase, and AI technology to compute expert-level answers. This functionality is made possible by the Wolfram Language. This article provides instructions on how to create a simple assistant application in Python that can answer basic questions.

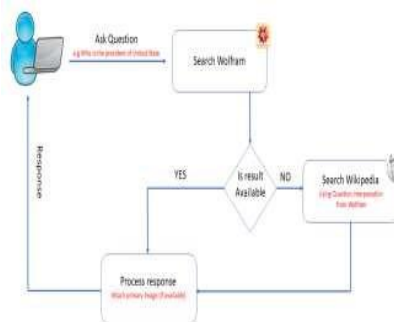


Figure 1

The user's input will be sent to Wolfram Alpha for processing. If a response is received, it will be returned to the user. If no response is received, the input will be interpreted as a keyword(s) and used for a Wikipedia search query.

III. EQUIPMENT – CHOICE OF MATERIALS

A. Tools and Environment Development:

- **Python 3:** Python is a high-level, object-oriented, interactive, and general-purpose programming language. Python 3.0 was introduced in 2008, which is incompatible with the earlier version, but several critical features have been backported to make it compatible with version 2.7.
- **Visual Studio Code:** Microsoft created a source-code editor that works on Windows, Linux, and macOS. It has debugging support, integrated Git control and GitHub, syntax highlighting, intelligent code completion, code snippets, and refactoring capabilities.
- **Sublime Text:** It is a shareware cross-platform source code editor that includes a Python API. It is compatible with many programming and markup languages, and plugins can be added by users, typically community-built and maintained under free-software licenses.

B. API and Python Libraries

- **Google Speech API:** Speech recognition is an essential function in numerous applications such as home automation, artificial intelligence, etc. This article introduces the use of Python's SpeechRecognition library. It is useful as it can be used on microcontrollers such as Raspberry Pi with an external microphone. We chose Google Speech API since it has an excellent Speech Recognition API. This API translates spoken text (microphone) into written text (Python strings), briefly known as Speech to Text. By speaking into a microphone, Google API converts it into written text. The API provides excellent results for the English language.
- **Wolfram Alpha API:** The API enables clients to submit free-form queries similar to those entered on the Wolfram|Alpha website, and the computed results are returned in various formats. It uses a standard REST protocol.

The pyttsx3 library in Python is a text-to-speech conversion tool. It operates offline and is compatible with both Python 2 and 3, making it a suitable choice over other libraries.

IV. CONCLUSION

The Project Development and Implementation

The program focuses mainly on developing Windows-based software using Python programming language, various APIs for Google products, and API for mathematical operations, among others. The project was developed by a team of three developers and follows the Incremental model and Extreme programming model. During the six-month development process, the team followed the same cycle in each phase of analyzing requirements, designing, implementing solutions in pair programming mode, and testing the results. The development process was guided by a primary planning strategy that defined how to work with the program, how much time to spend each week, the necessary resources for development, and how to handle problems that arose.

The project was efficiently completed under this development model, and the resources found early on were useful during implementation.

Project Usage & Potential Prospects

The project is highly useful and has significant potential for use in various industries. Although the program's primary focus is on creating a virtual assistant on Windows-based systems using voice, the concept of speech recognition can be applied in different industries. Speech recognition technology can be more convenient, save time, and be helpful, especially for people with manual operation difficulties. The program is a collection of 15-20 functions that are frequently used on a Windows PC, and users can enjoy different services within this platform. It is easy to use with simple operations compared to traditional working strategies where users must know how to work with desktop

systems. Additionally, the program's voice-operated feature is helpful for people who prefer voice operation and those who have difficulties/disabilities with manual operations. The primary objective of the program is to provide services using voice and make it accessible to more people.

REFERENCES

- [1]. S Subhash; Prajwal N Srivatsa; S Siddesh; A Ullas; B Santhos. Artificial Intelligence-based Voice Assistant, IEEE, 2020. <https://ieeexplore.ieee.org/document/9210344/>
- [2]. EV Polyakov, MS Mazhanov, AY Voskov, LS Kachalova, MV and SV Polyakov, "Investigation and development of the intelligent voice assistant for the IOT using machine learning", Moscow workshop on electronic technologies, 2018.
- [3]. Veton Kepuska and Gamal Bohota, "Next generation of virtual assistant (Microsoft Cortana Apple Siri Amazon Alexa and Google Home)", IEEE conference, 2018.
- [4]. Laura BURbach, Patrick Halbach, Nils Plettenberg, Johannes Nakayama, Matrina Ziefle and Andre Calero Valdez, "Ok google Hey Siri Alexa. Acceptance relevant of virtual voice assistants", International communication conference IEEE, 2019
- [5]. X. Lei, G. Tu, A. X. Liu, C. Li and T. Xie, "The insecurity of home digital voice assistants - amazon alexa as a case study", CoRR, vol. abs/1712.03327, 2017.
- [6]. J. Gratch, N. Wang, J. Gerten, E. Fast and R. Duffy, "Creating rapport with virtual agents" in Intelligent Virtual Agents, Berlin, Heidelberg:Springer Berlin Heidelberg, pp. 125-138, 2017.
- [7]. B. Weiss, I. Wechsung, C. Kühnel and S. Möller, "Evaluating embodied conversational agents in multimodal interfaces", Computational Cognitive Science, vol. 1, pp. 6, Aug 2015.
- [8]. Y. Matsuyama, A. Bhardwaj, R. Zhao, O. Romeo, S. Akoju and J. Cassell, "Socially-aware animated intelligent personal assistant agent", Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue, pp. 224-227, 2016.
- [9]. M. Schroeder, E. Bevacqua, R. Cowie, F. Eyben, H. Gunes, D. Heylen, et al., "Building autonomous sensitive artificial listeners", IEEE transactions on affective computing, vol. 3, pp. 165-183, 2018.
- [10]. Hannun, C. Case, J. Casper, B. Catanzaro, G. Diamos, E. Elsen, et al., "Deep speech: Scaling up end-to-end speech recognition", CoRR, vol. abs/1412.5567, 2014.
- [11]. M. Schröder and J. Trouvain, "The german text-to-speech synthesis system mary: A tool for research development and teaching", International Journal of Speech Technology, vol.
- [12]. P. Cosi, F. Tesser, R. Gretter, C. Avesani and M. Macon, "Festival speaks italian!" in EUROSPEECH 2001 Scandinavia 7th European Conference on Speech Communication and Technology 2nd INTERSPEECH Event, pp. 509-512, 2001.
- [13]. D. Huggins-Daines, M. Kumar, A. Chan, A. W. Black, M. Ravishankar and A. I. Rudnick, "Pocketsphinx: A free real-time continuous speech recognition system for hand-held devices" on Acoustics Speech and Signal Processing Proceedings, May 2018
- [14]. Think Python Book, Allen Downey
- [15]. A Python Book Author: Dave Kuhlman
- [16]. Python by Narayana Book