International Journal of Modern Developments in Engineering and Science
Volume 4, Issue 5, May 2025
https://www.ijmdes.com | ISSN (Online): 2583-3138

AI Based Voice Companion

Akshay Awasthi[1*], Karan Dwivedi[2], Nilesh Mani Tripathi[3], Vaishali Rastogi[4]
[1,2,3]Student, Department of Computer Science, Raj Kumar Goel Institute of Technology, Ghaziabad, India
[4]Assistant Professor, Department of Computer Science, Raj Kumar Goel Institute of Technology, Ghaziabad, India

***Abstract*: Voice assistants are software agents that can interpret human speech and respond via synthesized voices. Apple's Siri, Amazon's Alexa, Microsoft's Crotona, and Google's Assistant are the most popular voice assistants and are embedded in smart phones or dedicated home speakers. Users can ask their assistants questions, control home automation devices and media playback via voice, and manage other basic tasks such as email, to-do lists, and calendars with verbal commands. This paper will explore the basic workings and common features of today's voice assistants. Digital installation provides new opportunities to simplify everyday life using auxiliary technology tools. Amazon Alexa, Apple Siri, Microsoft Cortana, are examples of voice assistant. Voice assistant is a software that utilize artificial intelligence to take input in form of voice and then do the task accordingly. We use Various methods to convert speech into text (STT), then after processing of the text, it converts into speech (TTS). However, the smart personal assistant's study is vast. It is divided into separate branches, e.g. A computer-related environment, personal interaction with a computer as well Information systems. Various python packages are used in this project. In this paper, we have tried to present a descriptive and detailed review to provide strong support for future research. We use NLP instead of pattern identification strategies for recognizing contextual based text. It works online and offline. Python programming language is used in voice assistant. Data is stored in the app itself; it reduces the complexity of time and space.**

***Keywords*: Voice assistant, artificial intelligence, python.**

1. Introduction

Voice assistant is used to run a computer system like a laptop or PC at your own command. It is a software that can understand human's language and verbal commands and then performs according to the instruction provided to it. Users can give commands to their assistants. Voice assistant performs various functions like sending e-mail, managing to-do lists, open any websites or app, send texts with just any verbal commands on WhatsApp etc. Nowadays assistants are very helpful for human. It has brought comfort in human life. Now we do not have to type for searching anything on internet. We just have to tell the assistant verbally and voice assistant will understand your words and will do actions accordingly. Voice assistants take less time to show the results. We can save our time with the help of voice assistant. Voice assistants usually work with the internet, they require an Internet connection. To make a voice assistant we need to have knowledge of python language. Python has many inbuilt libraries and functions that makes coding part a bit easier. Virtual assistants are work oriented. Voice assistant is software that works according to the voice commands given to it. They can interpret and respond to human sounds. VS code editor is used to execute our python program. Some of the packages and modules of python that has been used are speech recognition, pyttsx3, smtplib, date time etc.

In recent years, with the gradual development of science and technology, people have changed the lifestyle of finding information in the library when they have a problem. Now people can directly know the answer to the question by using a search engine or artificial intelligence voice assistants. For example, when people use iPhone mobile phones, they often use the siri function. Users only need to activate Siri's voice assistant function and use some simple voice commands to query information, make calls, send information, get directions and play music. It can be seen that siri, as an artificial voice assistant, has completed the work of personal assistant to a certain extent. The data that used to be calculated by the human brain has been superseded by machines, and artificial intelligence is a machine that can imitation human responses. In fact, artificial intelligence voice assistants are based on applications, which often using voice commands to answer users' questions and perform tasks. This interaction is mainly achieved through voice assistants and automatic speech recognition systems that output a set of words or sentences for text by recognizing and transforming the input speech fragments.

2. Methodology

Voice assistants take voice input and perform tasks according to the instructions provided. Working of these assistants is very simple and easy. We give some instructions to assistant in the form of audio signal then the software understands and analyze those audio signal and after that it does the tasks. The python code which we have used is also not very complex. These voice assistants can also be helpful for the peoples who cannot see. Use of python language made the execution fast and simple also. Python code has some of installer packages like speech recognition, pyttsx3, python backend, system calls.

A. Text to Speech Module

Text-to-Speech (TTS) refers to the ability of computers to read text aloud. A TTS Engine converts written text to a phonemic representation, then converts the phonemic

*Corresponding author: satyamawasthi438@gmail.com

Table 1
Literature survey

Author	Research Objective	Methodology	Key Findings
Ayub et al. [7], (2019)	Performance analysis of blockchain-based data compression models for network optimization	Blockchain-based compression techniques, such as IPWR method	Blockchain integration significantly improves data compression and network performance, reducing latency and enhancing security.
Adomavicius et al. [10], (2005)	Deliver a comprehensive summary of cutting-edge blockchain-based data compression methods and explore future extensions	Blockchain, cryptographic techniques, and statistical methods	Blockchain enhances data compression by improving forecast accuracy and ensuring secure data transmission over networks.
Ahuja R et al. [12], (2019)	Propose a model using blockchain for secure and efficient data compression in network environments	Blockchain-based compression model, cryptographic techniques	Blockchain-based compression reduces data transfer times and lowers RMSE, improving network performance.
Koren Y et al. [15], (2010)	Analysis of the role of trust in blockchain-enabled data compression systems	Trust metrics in blockchain-based networks	Trust-based systems significantly improve data compression accuracy and efficiency in blockchain-enabled networks, especially with sparse data interactions.
Isinkaye et al. [1], (2015)	A comparison of blockchain-based data compression models and traditional methods	Blockchain-Data Integration, User-Item Association	Blockchain integration provides enhanced data security, reducing redundancy and ensuring efficient compression even with large data volumes.
Kassak et al. [4], (2016)	Hybrid blockchain and data compression strategies for optimized network throughput	Hybrid blockchain model for data compression	Hybrid blockchain-based compression outperforms traditional compression techniques by reducing data overhead and increasing throughput.

representation to waveforms that can be output as sound. TTS engines with different languages, dialects and specialized vocabularies are available through third-party publishers.

B. Python Backend

The python backend gets the output from the speech recognition module and then identifies whether the command or the speech output is an API Call and Context Extraction. The output is then sent back to the python backend to give the required output to the user.

C. Speech Recognition Module

Since we're creating a voice assistant app, one of the most critical features is that your assistant knows your voice. In the terminal, execute the following command to install this module.

i. *Python:* High-level, OOP-based language used in AI, ML, NLP, and data science. Supports rapid development with minimal code.
ii. *Que.py:* Transforms natural language into database queries with minimal coding.
iii. *Pyttsx3:* Offline text-to-speech library supporting Windows, macOS, and Linux.
iv. *NLP & Voice Recognition:* Enables speech recognition, intent detection, and command understanding.
v. *SQLite:* Lightweight, in-memory relational database included with Python; ideal for small to medium datasets.

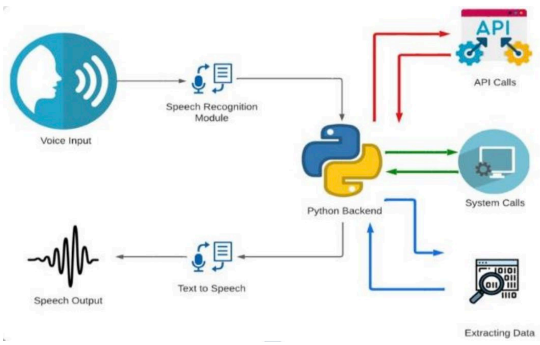

Fig. 1. System diagram

3. Conclusion

This paper presents a comprehensive overview of the design and development of a Static Voice enabled personal assistant for pc using Python programming language. This Voice enabled personal assistant, in today's lifestyle will be more effective in case of saving time and helpful to differently abled people, compared to that of previous days. This Assistant works properly to perform some tasks given by user. Furthermore, there are many things that this assistant is capable of doing, like sending message to user mobile, YouTube automation, gathering information from Wikipedia and Google, with just one voice command. Through this voice assistant, we have automated various services using a single line command. It eases most of the tasks of the user like searching the web etc., We aim to make this project a complete server assistant and make it smart enough to act as a replacement for a general server administration. The project is built using open-source software modules with PyCharm community backing which can accommodate any updates shortly. The modular nature of this project makes it more flexible and easier to add additional features without disturbing current system functionalities.

4. Future Scope

The future scope of AI-based voice assistants is vast and rapidly evolving. These systems are expected to develop stronger contextual understanding, enabling them to grasp user intent, emotions, and preferences more accurately. With advancements in multilingual processing, they will support a wider range of languages, accents, and dialects, making them more globally accessible. Integration with IoT will allow voice assistants to seamlessly control smart homes, vehicles, and industrial systems. Personalized user experiences will become standard, with assistants adapting responses based on individual habits and history. Security will improve through advanced voice biometrics and on-device processing, ensuring data privacy. Additionally, offline functionality will expand, making assistants more reliable in remote or low-connectivity areas. In fields like healthcare and education, voice assistants will play a

major role in patient support, remote diagnostics, and interactive learning. Future developments also include emotional intelligence, allowing voice assistants to respond empathetically based on the user's tone and mood.

References

[1] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. A., Kaiser, Ł., & Polosukhin, I. (2017). Attention is All You Need.

[2] Baevski, A., Zhou, H., & Auli, M. (2020). Wav2Vec 2.0: A framework for self-supervised learning of speech representations.

[3] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of deep bidirectional transformers for language understanding.

[4] Park, D. S., Chan, W., Zhan, J., & Le, Q. V. (2017). End-to-End Speech Recognition with Deep Convolutional Neural Networks.

[5] Various Authors. (2019). Real-Time Speech Recognition for Command Line Interfaces. Proceedings of the 2019 International Conference on Speech and Language Processing (SLP).

[6] Singh, K. B., & Zong, N. J. (2017). A Survey of Speech Recognition Technologies and Their Applications. Proceedings of the 2017 IEEE International Conference on Signal and Image Processing (ICSIP).

[7] Various Authors. (2018). Designing and Implementing Speech Interfaces for Command Line Tools.

[8] Baidu Research. (2014). Deep Speech: Scaling up end-to-end speech recognition.

[9] Shaughnessy, "Interacting with Computers by Voice: Automatic Speech Recognition and Synthesis" proceedings of the IEEE, vol. 91, no. 9, September 2003.

[10] Patrick Nguyen, Georg Heigold, Geoffrey Zweig, "Speech Recognition with Flat Direct Models", IEEE Journal of Selected Topics in Signal Processing, 2010.

[11] David L. Poole and Alan K. Mackworth, Mackworth, Python code for voice assistant: Foundations of Computational Agents.

[12] Nil Goksel, Canbek Mehmet, Emin Mutlu, "On the track of Artificial Intelligence: Learning with Intelligent Personal Assistant", proceedings of International Journal of Human Sciences, 2016.

[13] Keerthana S, Meghana H, Priyanka K, Sahana V. Rao, Ashwini B "Smart Home Using Internet of Things", proceedings of Perspectives in Communication, Embedded systems and signal processing, 2017.

[14] Sutar Shekhar, P. Sameer, Kamad Neha, Devkate Laxman, "An Intelligent Voice Assistant Using Android Platform", IJARCSMS, 2017.

[15] Rishabh Shah, Siddhant Lahoti, Lavanya. K, "An Intelligent Chatbot using Natural Language Processing", International Journal of Engineering Research, vol. 6, pp. 281-286, 2017.

[16] Luis Javier Rodríguez-Fuentes, Mikel Peñagarikano, AparoVarona, Germán Bordel, "GTTS-EHU Systems for the Albayzin 2018 Search on Speech Evaluation", proceedings of IberSPEECH, Barcelona, Spain, 2018.

[17] Ravivanshikumar, Sangpal, Tanvee, Gawand, Sahil Vaykar, "JARVIS: An interpretation of AIML with integration of gTTS and Python", proceedings of the 2019 2nd International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT), Kanpur, 2019.

[18] Veton Këpuska, "Next-Generation of Virtual Personal Assistants (Microsoft Cortana, Apple Siri, Amazon Alexa, and Google Home)," International Journal for Research in Applied Science & Engineering Technology, vol. 10, no. 4, pp. 2251-2253, April 2022.