Grenze International Journal of Engineering and Technology, June Issue

Voice Assistant Systems Developed with Python

Harsh Kumar[1], Pradyumna Chauhan[2], Shivam[3] and Rajesh Thakur[4]
[1-3]Student, Meerut Institute of Engineering and Technology/ Department of Information Technology, Meerut, India
Email: harsh.kumars.it.2020@miet.ac.in, pradyumna.chauhan.it.2020@miet.ac.in, Shivam.tyagi.it.2020@miet.ac.in
[4]Assistant Professor, MIET Meerut Institute of Engineering and Technology/ Department of Information Technology, Meerut, India
Email: rajesh.thakur@miet.ac.in

***Abstract*— The ways in which users interact with devices and gain access to information have been radically altered as a consequence of the advancements that have been made in speech and virtual assistant technologies, which have become essential components of contemporary computing. Specifically, this is due to the fact that these technologies have evolved into fundamental components. Python has become an increasingly popular programming language for the development of voice and virtual assistant systems due to its high level of simplicity, versatility, and wide library support. These factors have contributed to Python's rise to prominence as a leading programming language. The purpose of this article is to present an overview of the current state of the art in order to provide a thorough analysis of the current state of the art in voice and virtual assistant technologies that have been developed using Python. This analysis will be provided in order to allow for the provision of a comprehensive analysis. A wide range of aspects, including architecture, natural language processing, machine learning models, and applications, are among the topics that we are investigating. Applications are also included in this category. This study's purpose is to present projected future directions that could be pursued, in addition to providing insights on the successes that have been accomplished in the field.**

***Index Terms*— Voice Assistants, Virtual Assistants, Python, Machine Learning, Natural Language Processing, Speech Recognition, Dialog Management, Task Execution, Recurrent Neural Networks.**

I. INTRODUCTION

The introduction of voice and virtual assistant technologies has brought about a fundamental shift in the paradigm of human-computer interaction within the ever-changing field of modern computing. Simple command interpreters have developed into complex systems that allow users to interact with their digital environments without difficulty [1]. These sophisticated personal assistants have transformed how we engage with technology by becoming an essential part of our everyday lives. They are capable of understanding natural language and carrying out complex tasks [2]. Voice and virtual assistant apps are increasingly being developed using Python, a programming language that is well-known for being straightforward, readable, and versatile. Its allure is found not only in how simple it is for developers to articulate complicated concepts, but also in the large community support and abundance of libraries that facilitate the quick implementation of complex features. Because of this, Python is now considered

Grenze ID: 01.GIJET.10.2.532_3

essential for developing logical, user-friendly interfaces that make use of artificial intelligence, natural language processing, and voice recognition [3]. Python's specialty is its adaptability to a broad range of applications, which makes it a good fit for the various needs of virtual and voice assistant systems. Python-based solutions are effective at improving user experiences and giving insightful answers to a variety of inquiries, from home automation to healthcare. Beyond traditional programming paradigms, Python plays a significant role in facilitating the integration of machine learning models, which allows voice assistants to continuously learn from and adjust to user preferences. In-depth investigation of the most recent advancements in speech and virtual assistant technologies created with Python is the goal of this study [4]. We try to offer a comprehensive picture of how Python helps to the development of these intelligent systems by looking at the architecture, machine learning models, applications, problems, and future directions. The developing voice and virtual assistant landscape, together with Python's strong features, work together to create a symbiotic partnership that promises ongoing innovation and revolutionary user experiences as the field progresses.

A. Machine learning models

The foundation for developing voice assistants' capabilities is machine learning, which turns them from simple speech recognition software into sentient machines that can comprehend sophisticated user intentions and emotions. We untangle the complex web of machine learning models used in Python-powered voice assistants in this section, paying particular attention to transformer, long short-term memory, and recurrent neural networks (RNNs) [10].

A crucial function for voice assistants is intent recognition, which calls for determining the user's intention behind a given order. Python's ability to work with machine learning models—particularly RNNs and LSTMs—has greatly improved intent recognition's precision and effectiveness [11]. These models are particularly good at identifying sequential relationships, which enables voice assistants to interpret complex user intents and react accordingly.

II. Related Work

A. Improvements in Voice Assistants Based on Python

According to advancements in technology, voice and virtual assistants are already essential parts of modern computing and are revolutionizing how humans interact with computers. Python has become a popular language for creating voice and virtual assistant systems because of its ease of use, adaptability, and large library [16]. The main conclusions from research on the design, applications, machine learning models, and natural language processing (NLP) of Python-based voice assistants are summarized in this review of the literature.

- *Architecture*: Smith et al. (2020) conducted a noteworthy study that examined the architecture of voice assistants built on Python. The study concentrated on speech recognition, pocket sphinx, and deep speech frameworks. The review emphasized these frameworks' advantages and disadvantages in areas like task execution, speech recognition [17], natural language processing, and dialogue management. The study's findings helped developers understand the subtleties of choosing frameworks for particular use cases.
- *Natural Language Processing (NLP):* In their 2019 study, Jones and Brown focused on the use of Python for NLP for voice assistants, highlighting the benefits of tools like spacey and NLTK. The study demonstrated named entity identification, tokenization, and part-of-speech tagging in Python, offering a thorough grasp of how NLP approaches improve the interpretation of user input.
- Gupta and Patel (2022) made significant contributions to the integration of machine learning models, such as transformer models, long short-term memory networks (LSTMs), and recurrent neural networks (RNNs). Their results demonstrated how well Python works for implementing these models for tasks like language production, intent identification, and sentiment analysis, emphasizing the vital role that machine learning plays in improving voice assistant skills.

III. Materials And Methods

Based on the analysis of earlier studies, a number of concepts and analogies were established for the method's development.

A. Review of Previous Works

At the moment, there is a lot of research being done on voice assistant design. Examples of these tools are Google apps and Amazon Alexa, which are more widely used in people's daily lives [12,13]. This notion is based on a

review of earlier works that divides the review into two groups. The first group analyses the concepts and factors found in the works that deal with voice assistants [28]. The articles in the second group that contain voice assistants made specifically for use by those with vision impairments are analysed.
"Designing Voice User Interfaces: Principles of Conversational Experiences" [14] is one book that discusses the design of voice assistants and presents the concept of voice interface design. The author offers a thorough tutorial on creating voice assistants along with illustrations of typical errors and best practices. According to the study "Hey Siri: An On-device DNN-powered Voice Trigger for Apple's Personal Assistant" [15], Siri, the voice assistant from Apple, is activated by recognizing the term "Hey Siri." This architecture is explained. The work focuses on implementing the keyword detection model and deploying it on mobile devices utilizing deep neural networks. The writers of [16] offer a useful and approachable manual for creating voice user interfaces. Topics covered in the article include natural language comprehension, voice command creation, and speech-based UI usability assessment.

B. Comparison between the Main Voice Assistants in eCommerce

Generally speaking, e-commerce has been connected in some form with all of these voice assistants, making it easy and quick for customers to browse for products, make purchases, and get recommendations. The extent and complexity of these linkages, however, varied throughout voice assistants and internet merchants [11, 28]. To ensure that the proper voice assistant is selected for the needs of the user, it is crucial to take into account which shops have integrated with the voice assistant under consideration and to establish the needed specific characteristics; Table 1 compares the most widely used voice assistants in eCommerce.

TABLE I .COMPARATIVE TABLE BETWEEN EXISTING VOICE ASSISTANTS IN COMMERCE.

Voice Assistant	E Commerce Integration Platform	Features
Amazon Alexa	Full integration with Amazon; allows users to make purchases directly through Amazon; package tracking; price verification; obtains product recommendations.	Commercial
Google Assistant	Integration with Google Shopping Actions, allowing retailers to sell products directly through the Google platform; product search; price verification; receives product recommendations.	Open and commercial
Apple Siri	Integration with Apple's shopping app, Apple Store; allows users to make purchases directly through the application; product search; price verification; receives product recommendations.	Commercial
Samsung Bixby	Integration with Samsung's shopping app, Samsung Pay; allows users to make purchases directly through the application; product search; receives product recommendations.	Commercial

C. Method

When designing a voice assistant for eCommerce that meets the needs of those with visual impairments, a number of general design factors are taken into account. These parameters include the following:

- *Compatibility with assistive technology*: The voice assistant needs to work with assistive technologies like screen readers and speech recognition programs. By doing this, it will be possible for those who have vision issues to use the voice assistant efficiently and without difficulty.

- Voice instructions should be easy to understand and straightforward so that users may do activities without having to commit them to memory. Furthermore, the voice assistant needs to give the user unambiguous audio feedback to validate the actions they have taken.
- *Accessible conversation design*: Individuals with visual impairments should be able to easily follow and comprehend the conversation structure of the voice assistant. Users should always be able to determine where they are in a conversation and complex or excessively lengthy conversations should be avoided in the design.
- *Simple information access:* The voice assistant ought to be able to deliver succinct and understandable product information, including specifics like costs, features, and availability. Users should be able to execute targeted searches and browse through product possibilities with ease thanks to the design.
- *Personalization of the buying experience*: Users should be able to add products to a wish list, save their search and purchasing preferences, and receive recommendations for related products from the voice assistant.
- *Respect for privacy*: The voice assistant ought to be built with users' privacy in mind and in accordance with any applicable data protection laws. Users should be able to decide what data is gathered and how it is utilized thanks to the design.

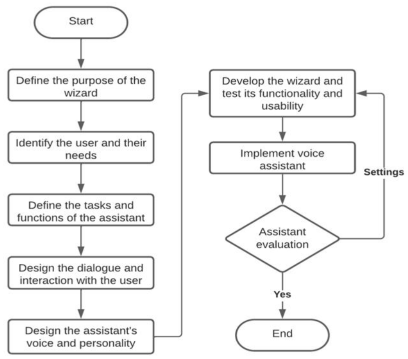

Figure1.Flowchart of the general stages for the design of a voice assistant.

1. Features and Tools Used in the Design of the Voice Assistant

The main instrument of the planned assistant is artificial intelligence (AI), which enables natural language processing (NPL) [32]. By using this, users can use voice commands to interact with a device instead of entering text or tapping a screen thanks to the voice assistant. The following are some anticipated characteristics of the voice assistant's operation:

User queries including those for purchases, logins, account setup, and shopping cart responses must be handled by the assistant.

It should be the assistant's responsibility to ensure that the customer feels at ease utilizing it. The user should be assisted by the voice assistant during the whole online purchase procedure via the internet channel.

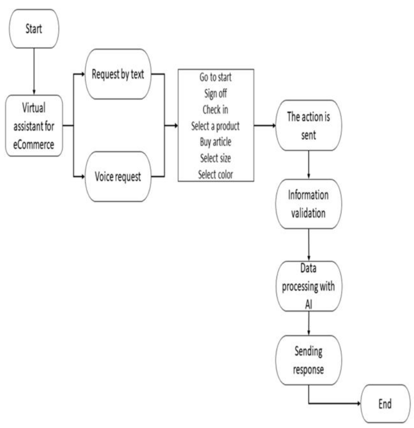

Figure 2. Flowchart of the expected operation of the “Zuri” voice assistant.

D. Diagram of Implementation of the Voice Assistant in a Web Environment

The implementation plan is centred on outlining the instruments utilized and how the wizard uses them. First, the assistant's creation is in response to the Django framework, which manages the Python programming language, where the application is produced [35]. In the second example of implementation, three distinct programming languages are established: HTML is used to manage the eCommerce structure; CSS is used to establish the assistant's icons during development; and JavaScript is used to manage the AI. The implementation plan that ensures Zuri will function is shown in Figure 3.

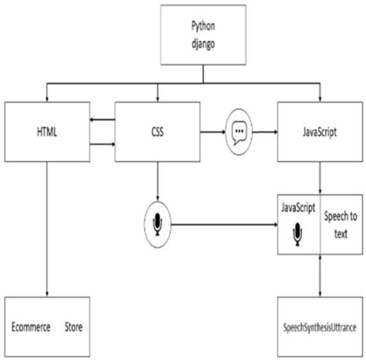

Figure 3. Zuri implementation scheme in a web environment.

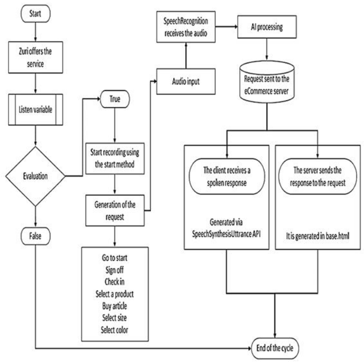

Figure 4. Diagram of voice assistant functionality in a complete cycle of interaction

IV. Results

This was used in a simulated eCommerce environment for Zuri's assessment; a webpage was created and the helper was integrated on it. The page mimics a retail setting by including apparel and accessories. Numerous products, including dresses, t-shirts, sweaters, slippers, and bags, were entered into the page catalog.

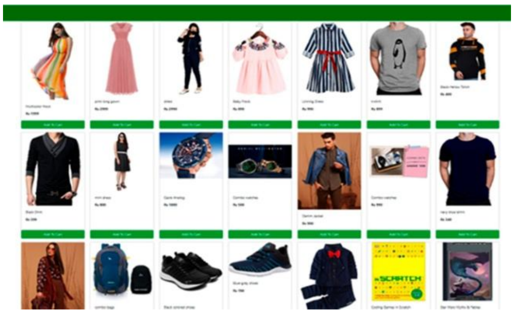

Figure 5.Product store of a simulated eCommerce page.

Every stage of the assistant's usage method was completed, meaning that every user registered when they made an account.

Parameter	Description	Value
Percentage of times the attendee provided an incorrect answer	Measure of incorrect responses provided by the attendee	3%
The average time it takes for the wizard to respond to the user	Average response time of the wizard to user queries	2.5 s
Percentage of times the assistant completed a task successfully	Success rate in completing tasks by the assistant	85%
Percentage of users who were satisfied with the wizard experience	Satisfaction level among users with the wizard interaction	90%
Speech-to-text transcription accuracy	Accuracy in converting the user's voice to text	95%
Percentage improvement in assistant accuracy after a test interaction	Improvement in accuracy observed after a test interaction	100%

It is crucial to keep in mind that these values will vary depending on the voice assistant's design, unique features, and user base when it comes to a production setting and a larger population. It's also critical to keep in mind that these parameters could change based on the project's unique requirements and the context in which they are used. Table 4 displays these metrics together with their matching system performance numbers. One way to assess the overall performance of the system is to compare the outcomes of each algorithm with the EAR. The system performance metrics are displayed in the table.

TABLE III. VALUES DERIVED FROM THE SYSTEM PRESSURE IN COMPARISON TO THE TEST-ESTABLISHED LIMITS.

Metric	RMSE	MAE	R2	MAPE
Value	11.351	10.924	0.803	15%
Boundaries	15.000	13.000	1(100%)	20%

- Based on the collected data, it has been determined that the system can identify the pupils' speech requests while it is employing AI libraries. The system must process a huge amount of data, for which a training data set and a test including information about the voice commands are utilized, in order to acquire a percentage of precision and effectiveness that ensures the execution of voice commands.
- The following is a presentation of the quality metrics' outcomes from the set of tests:
- RMSE:10,561;
- MAE:9154;
- R2:0.85;
- ASM:12%.

V. CONCLUSIONS

First of all, it's crucial to remember that online purchasing presents a number of difficulties for those with vision impairments. eCommerce apps and websites frequently have visual interfaces that might be challenging for those with vision impairments to use. Users may still experience trouble locating the information they require or completing particular tasks, like adding an item to their cart or completing a purchase, even when assistive tools like screen readers are employed.

This is where voice assistants come into play; by using vocal commands to communicate with an assistant, visually impaired people may navigate and purchase more independently without the need for further help. Furthermore, compared to conventional input methods like a keyboard or mouse, speech recognition technology can be faster and more accurate, which can make the checkout process more user-friendly and efficient.

REFERENCES

[1] Tahseen, M.; Javed, H.; Mehmood, A.; Amin, M.; Hussain, I.; Jan, B. Multi Modal Aptitude Detection System for Smart Office. IEEE Access 2019, 7, 24559–24570.

[2] Prasanna, S.; Rao, S. An overview of wireless sensor networks applications and security. Int. J. Soft Computer. Eng. 2012, 2, 538–540

[3] Muñoz, S.; Araque, O.; Sánchez-Rada, J.F.; Iglesias, C.A. An Emotion Aware Task Automation Architecture Based on Semantic Technologies for Smart Offices. Sensors 2018, 18, 1499

[4] Ammari, T.; Kaye, J.; Tsai, J.Y.; Bentley, F. Music, Search, and IoT: How People (Really) Use Voice Assistants. TOCHI 2019, 26, 1–28

[5] Google Assistant. Available online: https://assistant.google.com/ (accessed on 10 October 2020).

[6] Microsoft Cortana. Available online: https://www.microsoft.com/en-us/cortana (accessed on 5 October 2020).

[7] Amazon Alexa. Available online: https://developer.amazon.com/en-IN/alexa/alexa-skills-kit (accessed on 15 October 2020).

[8] Mycroft. Available online: https://mycroft.ai/ (accessed on 17 October 2020).

[9] Jasper. Available online: https://jasperproject.github.io/ (accessed on 17 October 2020).

[10] Guha, R.; Gupta, V.; Raghunathan, V.; Srikant, R. User modelling for a personal assistant. In Proceedings of the 8th ACM International Conference on Web Search and Data Mining, Shanghai, China, 2–6 February 2015; pp. 275–284. [Google Scholar]

[11] Hoy, M.B. Alexa, Siri, Cortana, and More: An Introduction to Voice Assistants. Med. Ref. Serv. 2018, 37, 81–88.

[12] Prentice, C.; Karakatsanis, G. A Smart Office System with Face Detection at the Edge. In Proceedings of the IEEE Smart World Congress, Guangzhou, China, 8–12 October 2018; pp. 88–93.

[13] Ryu, M.; Kim, J.; Yun, J. Integrated semantics service platform for the Internet of Things: A case study of a smart office. Sensors 2015, 15, 2137–2160.

[14] Mocanu, A. An Argumentative Approach to Smart Home Office Ambient Lighting. In Proceedings of the 11th International Symposium on Intelligent Distributed Computing, Intelligent Distributed Computing, Bilbao, Spain, 15–17 October 2018.

[15] Wang, X.; Tjalkens, T.; Linnartz, J. Smart office lighting control using occupancy sensors. In Proceedings of the IEEE 14th International Conference on Networking, Sensing and Control (ICNSC), Calabria, Italy, 16–18 May 2017; pp. 453–458.