INTERNATIONAL JOURNAL OF SCIENTIFIC RESEARCH IN ENGINEERING AND MANAGEMENT (IJSREM)
VOLUME: 07 ISSUE: 07 | JULY - 2023 SJIF RATING: 8.176 ISSN: 2582-3930

VOICE BASED VIRTUAL ASSISTANT

Kiran H, Girish Kumar, Hanumanta DH, Dilshad Ahmad, Lalitha S

Electronics and Communication Enginreering, BMS College of Engineering, Bangalore – 560019

Abstract— **The goal of this project is to create a voice-based smart virtual assistant that makes use of cutting-edge machine learning, speech recognition, and other technologies. Through personalized responses and recommendations, the main goal is to improve user experience and streamline tasks. Aside from setting reminders and booking appointments, the virtual assistant will also be able to make calls, send messages, and operate smart home appliances. It will also retrieve current data on the weather, news, sports scores, and other topics. The assistant will offer a seamless and intuitive interface across several languages and maintain contextual information from past talks with the help of context awareness and multilingual assistance. The virtual assistant will gradually enhance its effectiveness and gain a deeper understanding of user preferences by continuously learning from user interactions. An intelligent, hands-free virtual assistant that revolutionizes user interaction and task management is the goal of this project.**

Keywords— AI, Voice Assistant,

I. INTRODUCTION

The growth of AI over the last few decades has actively made it possible for both individuals and organizations to do the impossible. One of the most difficult advancements in computing has been giving machines the ability to think and judge. No matter what aspect of our lives we're trying to improve, AI can help—whether it's helping us read our emails, find our way around, or find new music or movies.

The Voice-Based Virtual Assistant (VBVA) has emerged as a revolutionary technology that combines artificial intelligence and voice recognition to provide a seamless and intuitive interaction between humans and machines. VBVA offers a hands-free and natural language interface that allows users to interact with a digital assistant through voice commands, enabling a wide range of applications and services.

A software application that employs artificial intelligence algorithms to comprehend and carry out voice commands is known as an AI voice assistant. These voice assistants may be found on many different gadgets, including smartphones, smart speakers, and even automobiles. Google Assistant, Sir, Microsoft Cortana, and Amazon Alexa are a few of the well-known artificial intelligence voice assistants. These AI voice assistants interpret voice instructions using natural language processing (NLP) and machine learning algorithms, then provide precise and useful information in their responses. Over time, they continue to develop and get better, becoming more intelligent and able to handle a greater variety of jobs.

Several problems and limitations may arise in the absence of Voice-Based Virtual Assistant (VBVA).

Manual Input Overload: Without a VBVA, users would rely on manual input methods such as typing or tapping on screens to interact with digital systems. This can be time-consuming and cumbersome, particularly for tasks that require frequent or complex input.

Lack of Hands-free Interaction: Users would be unable to perform tasks or access information in a hands-free manner. This limitation can be particularly challenging in situations where manual interaction is impractical or unsafe, such as when driving or performing physical activities.

Limited Accessibility: Users with physical disabilities or impairments may face barriers in accessing and interacting with digital systems that heavily rely on manual input methods. The absence of voice-based interaction options reduces accessibility and inclusivity for these individuals.

Reduced Efficiency: Manual input methods are generally slower compared to voice-based interactions. Users may experience decreased efficiency and productivity, especially when dealing with tasks that involve multiple steps or complex inputs.

Limited Multitasking: Without a VBVA, users may find it challenging to multitask effectively. They would need to switch between different applications or devices manually, resulting in interruptions and decreased efficiency.

Missed Opportunities for Personalization: Voice-based interaction enables virtual assistants to gather information about user preferences, behaviors, and needs. Without a VBVA, the ability to provide personalized recommendations or tailored assistance may be limited.

II. PROPOSED SOLUTION

The Voice-Based Virtual Assistant (VBVA) project has the potential to solve various problems and offer several benefits to users.

Hands-free and Convenient Interaction: The VBVA allows users to interact with digital systems and devices using voice commands, eliminating the need for manual input. This hands-free interaction provides convenience, especially in situations where manual interaction is difficult or not possible, such as while driving, cooking, or when physically impaired.

Personal Assistance: The VBVA can act as a personal assistant, helping users with tasks such as setting reminders, scheduling appointments, managing to-do lists, and providing personalized recommendations. It simplifies daily activities and improves productivity by assisting users in organizing their tasks and accessing relevant information effortlessly.

Information Retrieval: The VBVA can quickly retrieve information from various sources, including the internet, databases, and external services. Users can ask questions or seek specific information, such as news updates, weather forecasts, sports scores, or general knowledge. The assistant provides instant answers, saving users time and effort in searching for information manually.

Task Automation: The VBVA can automate repetitive tasks, such as sending messages, making phone calls, or controlling smart home devices. By simply issuing voice commands, users can perform these tasks more efficiently, streamlining their daily routines and freeing up time for other activities.

Accessibility and Inclusivity: The VBVA enhances accessibility for individuals with disabilities by offering a voice-based interface that eliminates barriers posed by traditional input methods. People with motor disabilities or visual impairments can interact with digital systems more easily, empowering them with greater independence and inclusivity.

Customer Support and Service: The VBVA can handle customer queries, provide support, and offer assistance in various industries such as e-commerce, banking, and telecommunications. By automating customer support tasks, it can improve response times, provide accurate information, and enhance overall customer satisfaction.

Smart Home Control: The VBVA can integrate with smart home devices and systems, allowing users to control lighting, temperature, security systems, and other connected devices through voice commands. This simplifies home automation and makes the management of smart home technologies more intuitive and user-friendly.

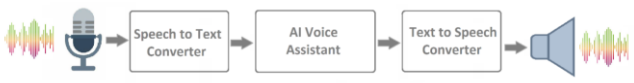

Fig 1: Block diagram of Voice based smart virtual assistant

A. Speech to Text Converter:

The primary tool in the Speech to Text converter is a microphone. This might be a phone, laptop, or specialised microphone. We are employing a Python module named "Speech Recognition" for speech recognition and text conversion.

A well-known Python package for carrying out voice recognition tasks is called Speech Recognition. It is a straightforward, user-friendly library that offers a standard interface for interacting with a variety of speech recognition tools, including Google Speech Recognition, CMU Sphinx, and many others. Speech may be recognised by the library from both microphone inputs and audio files.

B. AI Voice Assistant:

There are a number of AI-powered voice assistants available, but we are employing the ChatGPT, the most well-known AI system, whose API is offered by OpenAI.
ChatGPT API was just released by OpenAI. This API uses the same model as the ChatGPT product, which is gpt-3.5-turbo.
An AI chatbot called ChatGPT (Generative Pre-trained Transformer) was initially developed using a group of large language models (LLMs) referred to as GPT-3. The availability of OpenAI's newest GPT-4 models has just been revealed. For instance, ChatGPT's most innovative GPT-3.5 model was trained using 570GB of internet-sourced text data, which according to OpenAI contained texts from books, journals, websites, and even social media.

C. Text to Speech Converter:

Speaker is the key element in text to speech conversion. For Text to Speech Conversion, we are utilising the well-known Python package "pyttsx3".

A Python text-to-speech conversion library is called pyttsx3. It is compatible with Python 2 and 3 and works offline, unlike competing libraries. To obtain a reference to a pyttsx3. Engine instance, an application calls the pyttsx3.init() factory method. It is a very user-friendly programme that turns typed text into speech. The "sapi5" for Windows programme offers two voices that the pyttsx3 module supports: a female voice and a male voice.

III. LITERATURE SURVEY

1). “Voice Control Using AI-Based Voice Assistant” by S.Subhash et al., (2020)
This paper shows the usage of Voice to Automate and Control things using Artificial Intelligence. The voice assistant will gather the audio from the microphone and then convert that into text, later it is sent through GTTS (Google text to speech). GTTS engine will convert text into audio file in English language, then that audio is played using play sound package of python programming Language. have built an AI-based voice assistant which can do all of these tasks without inconvenience. They have created a function, Intelligent Personal Assistant which can perform mental tasks like turning on/off smart phone applications with the help of Voice User interface (VUI) which is used to listen and process audio commands.

2). “Voice-Based Human Identification using Machine Learning” by A. Alsaify et al., (2022)
This paper proposes a methodology for speaker recognition based on machine learning algorithms. The authors used Support Vector Machine (SVM) and Random Forest (RF)

models with statistical features and Mel-Frequency Cepstral Coefficients (MFCC) as the input features of the models. A new voice dataset was collected for the purpose of training and evaluating speaker recognition models. Samples were obtained from non-native English speakers from the Arab region over the course of two months. The performed experiments showed that using the developed methodology and the collected dataset, a 94% identification accuracy can be achieved.

3). "Voice-based person identification using mean clustering and medoid clustering approaches" by Anil B V et al.,
This paper presents a study on voice-based person identification using mean clustering and medoid clustering approaches. The authors have implemented three classes of c-means: Hard c-means, fuzzy c-means, and rough c-means algorithms and compared them with three classes of medoid algorithms: hard c-medoid, fuzzy c-medoids, and rough c-medoid. The fuzzy c-medoid algorithm outperforms the other voice-recognition systems in terms of recognition accuracy. Well-grounded biometric recognition systems are of maximum confidence level to authenticate individuals and minimize the chances of security threats. Possessions of validating articles like identity card and/or memorized entities such as PIN or password usually performs the best. However, losing of the article or forgetting the entities can be really a catastrophic. Measuring the two categories of features namely physical or physiological with calculated precision leads to biometrics and identification of the person with biometric attributes leads to biometric recognition. A biometric system is essentially a pattern recognition systemthat measures biometric data from an individual. The estimated biometric data compared against the biometric data template stored in the biometric feature database. Recognition by voice addresses this problem with its merits related to Universality, Permanence, Collectability, and Acceptability.

4). "Enabling Robots to Understand Incomplete Natural Language Instructions Using Commonsense Reasoning" by Haonan Chen et al., (2020)

This paper proposes a new method called Language-Model-based Commonsense Reasoning (LMCR) that enables a robot to listen to a natural language instruction from a human, observe the environment around it, and automatically fill in information missing from the instruction using environmental context and a new commonsense reasoning approach1. The paper was presented at the 2020 IEEE International Conference on Robotics and Automation (ICRA). Natural language is inherently unstructured and often reliant on common sense to understand, which makes it challenging for robots to correctly and precisely interpret natural language. Consider a scenario in a home setting in which a robot is holding a bottle of water and there are scissors, a plate, some bell peppers, and a cup on a table. A human gives an instruction, "pour me some water", to the robot. This instruction is incomplete from the robot's perspective since it does not specify where the water should be poured, but for a human, it might be obvious that the water should be poured into the cup. A robot that has the common sense to automatically resolve such incompleteness in natural language instructions, just as humans do intuitively, will allow humans to interact with it more naturally and increase its overall usefulness. Language-Model-based Commonsense Reasoning (LMCR), a new approach which enables a robot to listen to a natural language instruction from a human, observe the environment around it, automatically resolve missing information in the instruction, and then autonomously perform the specified task.

5). "Joint Contextual Modeling for ASR Correction and Language Understanding" by Yue Weng et al., (2020)
This paper proposes multi-task neural approaches to perform contextual language correction on ASR outputs jointly with LU to improve the performance of both tasks simultaneously. The authors used a public benchmark, the 2nd Dialogue State Tracking (DSTC2) corpus, to measure the effectiveness of this approach. They trained task-specific Statistical Language Models (SLM) and fine-tuned state-of-the-art Generative Pre-training (GPT) Language Model to re-rank the n-best ASR hypotheses, followed by a model to identify the dialog act and slots. They further trained ranker models using GPT and Hierarchical CNN-RNN models with discriminatory losses to detect the best output given n-best hypotheses. They extended these ranker models to first select the best ASR output and then identify the dialogue act and slots in an end-to-end fashion. They also proposed a novel joint ASR error correction and LU model, a word confusion pointer network (WCN-Ptr) with multihead self-attention on top, which consumes the word confusions populated from the n-best. They show that the error rates of off-the-shelf ASR and following LU systems can be reduced significantly by 14% relative with joint models trained using small amounts of in-domain data.

IV. RESULTS AND DISCUSSION

HAI (Home Artificial Intelligence) system is an application that can run on Windows, Android, and iOS. It is basically a smart virtual assistant system that interacts with the user via the voice and text. The HAI system can listen to user's audio via the inbuilt system microphone and does the internal processing to convert the Audio to Text format, the Text is then given to the OpenAI via the API call. The response from the OpenAI is displayed on the screen and also converted into Speech.

Fig 2: UI

The HAI Application has various pages like, Chat, To-Do List, Reminders and Settings, these can be seen in Fig 3

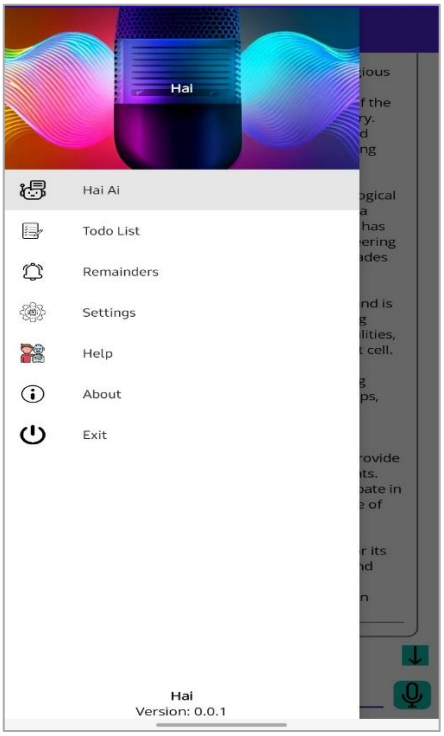

Fig 3: HAI app Menu window

The opens by default on the HAI Chat page. This page provides an interface to interact with the Open AI ChatGPT model.

The user has 2 ways of interacting with the HAI system.

Using Keyboard: This is the default way to interact with the HAI system. User can simply type in the type window. User should then press the send button, which will send the user typed data to the OpenAI ChatGPT model via the API call. The HAI system will then display the ChatGPT response.

Using Microphone: To use this method of Input, user has to press the Mic Logo on the bottom right corner, The HAI system will then listen to User's Speech, converts it into text, feed text into Open AI ChatGPT model, gets the text output from the ChatGPT model and displays in on the window and will also reads out the text.

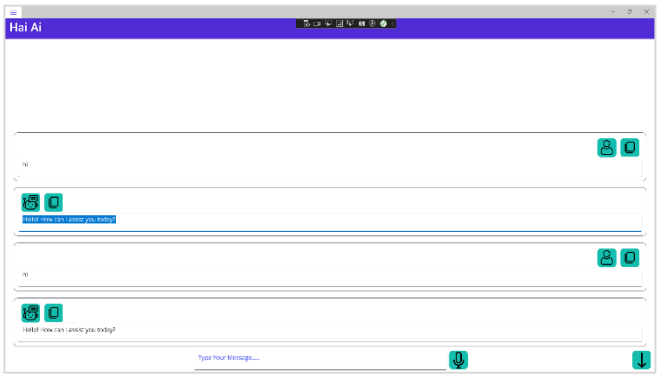

Fig 4: HAI Application Interface on Desktop

Fig 4 shows the HAI Application running on Windows Computer. The functionality and working are same as the android application.
The user can interact with the system suing either Keyboard or Mic. The response from the Chat-GPT are displayed.

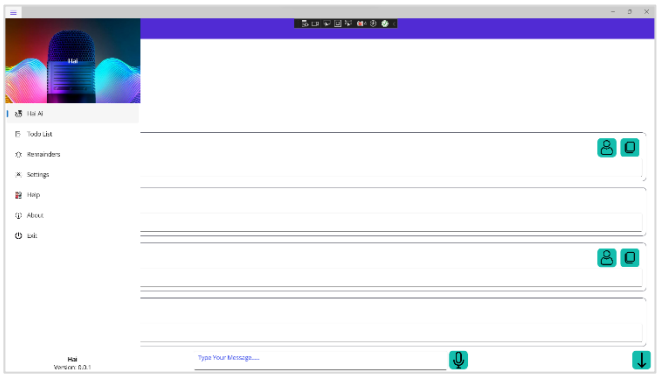

Fig 5: HAI App Menu Window on Desktop

Fig 5 shows the HAI App Menu Window. The Menu has HAI AI, To-Do List, Remainders, Settings, Help, About and Exit buttons. Currently the app version is 0.0.1.

User can interact with AI using HAI AI page, User can set tasks in To-Do list page, User can set the remainder using the Remainders page, User can find the app settings under Settings page, User can find the help related to the app under Help page, User can get the Application information under About page, User can exit the application using the Exit Button.

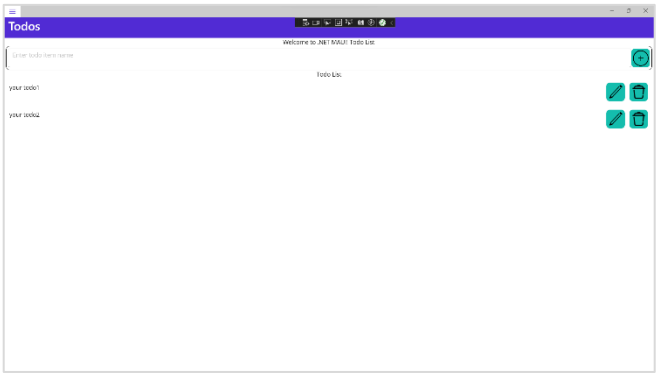

Fig 6: HAI App To-Do List

Fig 6 shows the To-Do List functionality of the HAI app. To-Do Task or List can be added by using the button with + icon.

Then using the pencil icon button, user can add the To-Do tasks, users can delete the tasks using delete button, User can save the task using the save button.

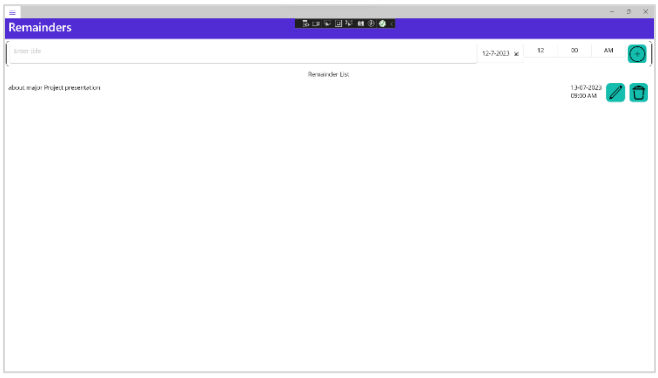

Fig 7: Hai App Remainders

Any remainders can be added to the app by pressing the + icon in remainders tab. User can then type the text that they want to be reminded of, user can also select the date and time to get the remainder. If user wish to delete the remainder, they can do so by using the delete button on the same tab.

V. CONCLUSION

The project's goal is to create a clever virtual assistant with AI algorithms that can recognise and react to voice commands. The objective is to give users a more comfortable and natural manner to interact with technology. Users might be forced to complete activities manually without a voice-based smart virtual assistant, which can be laborious and ineffective. They can also miss out on the advantages of using a technological interface that feels more natural and intuitive.

Users can execute tasks hands-free and without the use of a keyboard or touchscreen with the aid of voice-based smart virtual assistants. Additionally, they can speed up processes, increase accessibility, and improve the user experience in general. In order to understand voice instructions and provide precise and useful information in response, the project will use natural language processing (NLP) and machine learning methods. Users will save time and effort by using the virtual assistant to complete things like setting reminders, making calls, or placing orders for products.

VI. FUTURE WORK

The Voice Based Smart Virtual Assistant is currently implemented to be used on Windows, Android and iOS. Further this project can be implemented using Raspberry Pi or any compact processing unit with dedicated Mic and Speaker. Further we can implement Home Automation like Controlling Light, AC, etc.

VII. REFERENCES

1. S. Subhash et al., “Voice Control Using AI-Based Voice Assistant,” 2020 International Conference on Smart Electronics and Communication (ICOSEC), Bangalore, India, 2020, pp. 592-596, doi: 10.1109/ICOSEC49019.2020.9230327.
2. B. A. Alsaify, H. S. Abu Arja, B. Y. Maayah, M. M. Al-Taweel, R. Alazrai and M. I. Daoud, "Voice-Based Human Identification using Machine Learning," 2022 13th International Conference on Information and Communication Systems (ICICS), Irbid, Jordan, 2022, pp. 205-208, doi: 10.1109/ICICS55353.2022.9811154.
3. A. B. V and R. M. S, "Voice Based Person Identification Using c-Mean and c-Medoid Clustering Techniques," 2020 IEEE International Conference on Distributed Computing, VLSI, Electrical Circuits and Robotics (DISCOVER), Udupi, India, 2020, pp. 121-126, doi: 10.1109/DISCOVER50404.2020.9278042.
4. H. Chen, H. Tan, A. Kuntz, M. Bansal and R. Alterovitz, "Enabling Robots to Understand Incomplete Natural Language Instructions Using Commonsense Reasoning," 2020 IEEE International Conference on Robotics and Automation (ICRA), Paris, France, 2020, pp. 1963-1969, doi: 10.1109/ICRA40945.2020.9197315.
5. Y. Weng et al., "Joint Contextual Modeling for ASR Correction and Language Understanding," ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Barcelona, Spain, 2020, pp. 6349-6353, doi: 10.1109/ICASSP40776.2020.9053213.