|| Volume 6 || Issue 6 || June 2021 || ISSN (Online) 2456-0774
INTERNATIONAL JOURNAL OF ADVANCE SCIENTIFIC RESEARCH AND ENGINEERING TRENDS

HUMAN INTELLIGENT DIGITAL ASSISTANT YOUTH : VIRTUAL DESKTOP HANDLER

Aishwarya Gaikwad[1], Sushant Parab[2], Ajay Bharsakale[3],Varad Deshpande[4], Pooja Ghodke[5], Jyoti Chauhan[6]

Computer Engineering, G.H.Raisoni Institute of Engineering and Technology, Pune 412207, Maharashtra, India.[1,2,3,4,5,6]

--- ***---

Abstract: We all refer to doing our task intelligently, swiftly, and without spending a lot of time in the twenty-first century. We are now all entering a new world, a digital world. We are all surrounded by digital technology. Today, one of the most common approaches, known as "Virtual Assistance," is gaining popularity. So, by providing software for decreasing work load and increasing speed, both normal and abnormal individuals will be able to benefit from it. With no further expertise required, it is extremely simple to use and dependable. It is entirely software-based in this case. The usage of external devices is optional (mouse, keyboard, headsets, etc).

Keywords- *Deep Learning, Speech recognition, Modules, Python language.*

--- ***---

I INTRODUCTION

<u>Introduction</u>- The great expansion in technology has increased the importance of virtual help, which is now one of the most popular and widely utilized in daily life. Handling the desktop through speech is an example of how virtual assistance may help you succeed. This is the software foundation, and it plays a part in completely voice-operating the desktop. It also focuses on the handicapped as well as the ignorant. The fundamental objective is to create a platform that is self-contained, adaptable, and simple to use.

This programme is based on the Machine Learning idea. The Python language is used to construct the programme, which is built using the Visual Studio Code platform. As we all know, the virtual help provided by the desktop handler may be employed both indoors and outside.

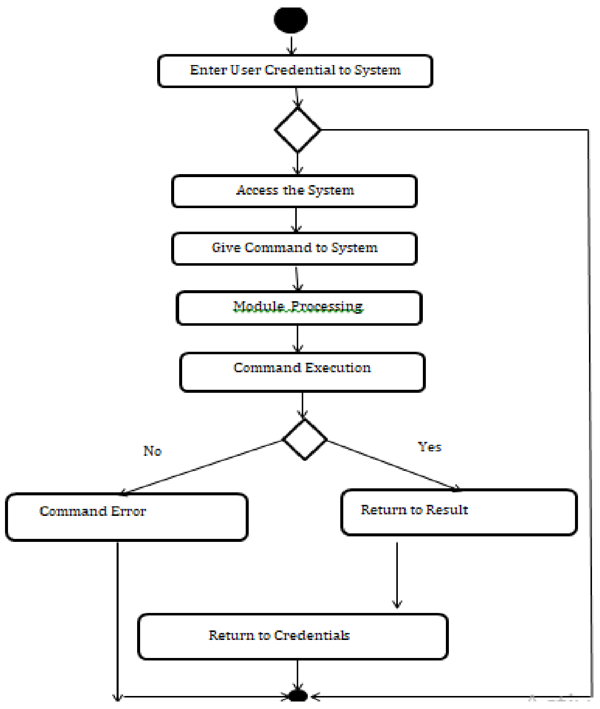
We've seen a lot of virtual help on the market, for example, Google and so on, each of which does a certain activity, but this programme is designed to perform all tasks and not just one.

This programme can present the answers to the questions that have been asked in an organised manner, guiding users (clients) via speech, or optionally utilising external devices such as a keyboard, mouse, or tactile input to detect the voice communicate with the machine and perform action as quickly as feasible. It was created with the use of deep learning and artificial intelligence Dialog flow as a simple approach to produce and recognise words.

All typing-related tasks, such as typing in notepad, Microsoft Office, and other applications, are handled by this programme. The programme may be launched and dismissed with the use of voice commands with this programme. The unique feature of this programme is that it allows you to utilise voice or external devices such as a mouse, keyboard, or headphones at any moment.

People with disabilities (handicaps) can use it very easily, and it can help them get jobs in data entry or as a presenter, for example. People who are uneducated or undereducated can also handle it flexibly, as it is a system that can also answer questions asked by the individual, and it is inexpensive.

We created interactive material for an introductory undergraduate course on Deep learning using the software Virtual Assistance for Desktop, a virtual assistant that allows normal and abnormal individuals (for example, individuals with disabilities) to access interactive information. This programme can display answers to frequently asked questions in a hierarchical organised way, prompting the user (client) to interact with the voice using either speech or external devices such as a keyboard, mouse, or tactile input.



Model of natural language Another benefit of this platform is its capacity to capture use data that might be helpful for personal purposes in areas outside of the home, such as industrial areas. The major goal of this programme is to explain the approach that drove our implementation so that it may be replicated in various everyday life situations, as well as to research Virtual Assistance for Desktop software as a means of advancing in the digital world. Several articles, news stories, and blogs are now discussing the potential, implementation, and effect of software, for example: In general, chat boards exist, but there is little to no literature suggesting a methodology for reproducing them in the real world as well as outside the door (example: educational field, industrial, ETC). In this regard, we created four primary categories as a generic structure for course material, with an emphasis on ease of implementation, updating, and generalisation. The finished work garnered positive feedback.

II LITERATURE REVIEW-

The goal of this programme is to create a useful and technological platform for everyone.

Intelligent virtual assistants (IVAs) are used in this Virtual Assistance software to improve the customer experience by providing immediate, smart, and efficient self-service. It is always available to provide support, answer questions, and take action on behalf of the user for a better user experience, lower costs, and optimised service channels. It also provides assistance to individuals and groups. Also in this software the PyAudio is the model that it can provide Python bindings for Port Audio, the cross-platform audio I/O library.

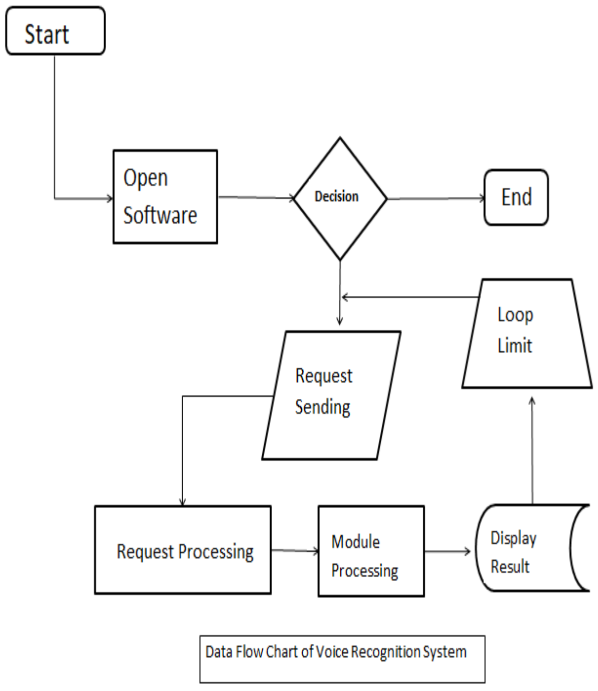


Figure 1: Working Design of the Virtual Assistance for Desktop

Python is the programming language used to create this app. It is one of the most widely used as well as independent languages, as well as the most recent and dependable. To make the software practical, the Visual Studio Code platform was utilised. All about Visual Studio Code may be found in this database.

It was created with the use of deep learning and artificial intelligence Dialog flow as a simple approach to build and identify a natural language model. Another advantage of this platform is its capacity to collect usage statistics in public areas, such as industrial areas, which might be beneficial for personal purposes. This software's principal goal is to lay forth the approach that drove our discourse.

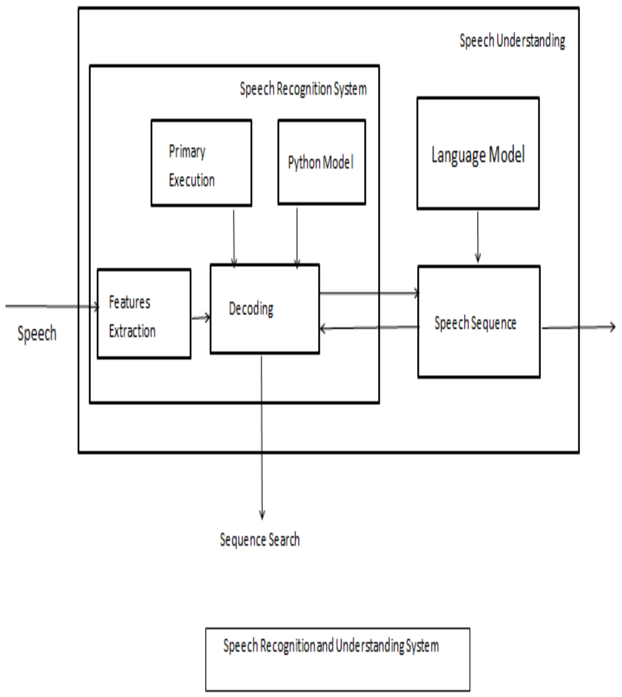


Fig: Speech Recognize Understating System

The user due to its accessibility and well structure data and voice recognize.

Step 1: Audio Processing

In step 1, the user talks into the device's microphone, which records an audio file. The audio file is a collection of numbers that contains data about the audio stream. The audio file may contain features that speech recognition engines utilising IPA OR IVA cannot process, depending on the recording equipment.

Step 2: Speech Recognition Engine

The Language Model (LM), Acoustic Model (AM), and Dictionary are required for the IVA OR IPV architecture to function. The dictionary is nothing more than a collection of words. The LM is a phonetic breakdown of the Dictionary's words; it comprises the phonetic units (phonemes) that make up those words. A list of acceptable words and their phonetic units may be found in LM. The AM is a file containing probabilistic values that are used to identify phonemes from audio file values. The AM is used to distinguish between speech phonetic units.

Step 3: Generalizer

The IPA may be multilingual thanks to the Generalizer. It's a function that takes each word from step (3) and returns a token or a null value for each one. A term that is a more "generic definition" of the entered word is an outputted token. e.g.

Generalize (“start”) = “ON”.

Here, the input word: “start”, is generalized to the output token: “ON”, which is a more general definition of “start”. The Generalizer is important because there are multiple words (usually synonyms of each other), which imply the same meaning in a certain context. e.g., similar to “start” -

Generalize (“open”) = “ON”

Generalize (“on”) = “ON”

Step 4: Finite State Automata

In the deterministic finite state automaton, traversal between the current state and an adjacent state can only take place – if the “required token” of one of the outgoing edges of the current state equals the “input token”. When a “sequence of tokens”, from diagram 3, is inputted to the FSA – depending on the sequence, the FSA is traversed and a particular end state is reached - and if the end state is an accepting state, the FSA sends instructions to the next step to perform an action depending on the specific end state.

We have designed our FSA to perform actions in response to the user’s commands. In step (6), the Generalizer deduces that what the user is saying - and based on that information in step (7) the FSA deduces what the IPA needs to do. Our design of the FSA is shown in Fig.3.

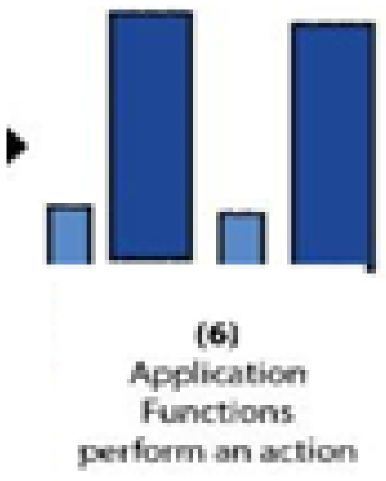


Figure 3: Application Function Diagram

Step 5: Application Functionality

The application functionality is the software that conducts observable activities for the user on the user interface (which might be visual or aural), or may either observe or receive responses/errors from the IPA.

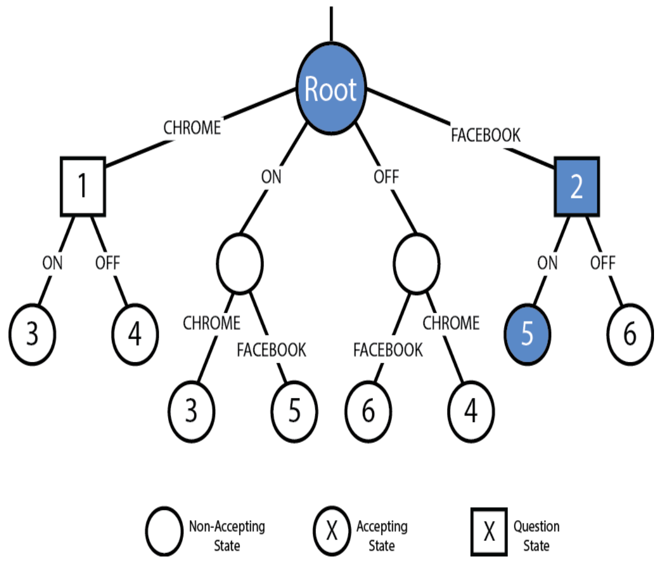


Figure 4: The design of the Finite State Automaton

Step 5: Application Functionality

The application functionality is the software that conducts observable activities for the user on the user interface (which might be visual or aural), or may either observe or receive responses/errors from the IPA.

CONCLUSION-

This software, particularly the typing application, will be the future development trend of smart software desktop handling via voice trends to smart development. In this study, we use the Intelligence Virtual Assistant (IVA) or Intelligence Personal Assistant (IPA) software agent to construct a smart and dependable desktop/pc virtual assistant based on deep learning that can execute tasks or services. This is a thorough examination of speech Racemization, emotion Racemization, and Indian/US English word vectors and applications. This research can reach the highest level of accuracy in a virtual assistant for brief sentences. External gadgets might be desired or not desired. (For example, a keyboard, mouse, and headphones). Combining a virtual assistant with a personal desktop in the future may result in smart software for desktop handlers.

REFERENCES-

1) Received March 8, 2020, accepted March 26, 2020, date of publication March 30, 2020, date of current version April 14, 2020.Digital Object Identifier 10.1109/ACCESS.2020.2984383.

2) Received September 13, 2019, accepted October 1, 2019, date of publication October 7, 2019, date of current version November 1, 2019.Digital Object Identifier 10.1109/ACCESS.2019.2945791

3) Received June 6, 2020, accepted June 25, 2020, date of publication June 29, 2020, date of current version July 8, 2020.Digital Object Identifier 10.1109/ACCESS.2020.3005733.

4) Received August 27, 2020, accepted September 16, 2020, date of publication September 22, 2020, date of current version October 6, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3025941
5) Received July 19, 2017, accepted August 21, 2017, date of publication August 31, 2017, date of current version September 27, 2017. Digital Object Identifier 10.1109/ACCESS.2017.2747626
6) Received July 6, 2017, accepted July 29, 2017, date of publication August 9, 2017, date of current version October 12, 2017. Digital Object Identifier 10.1109/ACCESS.2017.2737655
7) Received December 22, 2019, accepted January 9, 2020, date of publication February 12, 2020, date of current version March 23, 2020. Digital Object Identifier 10.1109/ACCESS.2020.2973373