Voice Based System Assistant using NLP and deep learning

K. Amaravathi [1], K. Subba Reddy[2], K. Sesha Shai Datta[3], A. Tarun[4], S. Ajay Varma[5]

[1] Assistant Professor, Department Of Computer Science & Engineering, Anil Neerukonda Institute Of Technology & Sciences, Visakhapatnam, Andhra Pradesh, India.

[2,3,4,5] Students, Department Of Computer Science & Engineering. Anil Neerukonda Institute Of Technology & Sciences, Visakhapatnam, Andhra Pradesh, India.

Abstract:

Desktop assistants also evolved during the period along with the humans. Nowadays people are very used to it and it is a part of their day-to-day lives.

We all want to make using computers more pleasurable. So there is one way by using an assistant. The typical path to provide instruction to the computer is through the keyboard, but there are more efficient ways to do it. Voice instruction is a more convenient way to give the command.

These systems were used in many different applications like human-computer interactions. It plays an important role in some people's lives, like the physically disabled. Some people may find it hard to use the system, so to overcome those kinds of issues we use virtual assistance that helps in their daily life. This Paper Builds a general purpose that makes conversations between user and computer.

Keywords:

Python+, deep learning, speech recognition, PQt5, NLTK, Numpy, TensorFlow, Keras, sequential model, tokenization, GUI, lemmatization, QWidgets.

1. Introduction:

Machines had to learn how to hear, recognize, and analyze human speech. long before the voice assistants integrated into our smart speakers could understand our requests to play music, switch off lights, and read the weather report. The technology we use today has been under development for more than a century, and it has come a long way from the first listening and recording equipment. From the phonograph to the virtual assistants. Assistant is described as one of the most advanced and promising interactions between humans and machines. An Assistant is computer software that can perform a conversation with a user in natural language through applications, mobile apps, or desktop applications. Different APIs are used by different firms, such as Google and Apple. It is absolutely remarkable that one can now organize meetings or send emails using only spoken commands. Whenever we want to perform an action on the system then we have to

communicate with it using our hands. If a person is facing difficulty communicating with the system physically, then it will not be proper communication. To overcome these kinds of problems we made an initiative to develop an application. This encompasses Braille systems, screen magnifiers, and screen reader development. Recently, efforts have been made to build tools and technology to assist blind people in using the internet.[3]. Anyone, able or unable, will be able to use such applications effectively. Unlike systems that focus on regular users' GUI friendliness, our approach meets the needs of both normal and visually and physically handicapped users.[6]. With the rapid development of deep learning techniques, it is now possible to solve these types of complicated problems utilizing neural networks, which overcomes the constraints of traditional machine learning methodology. We can extract high-level features from provided data using deep learning and neural networks. In this way, the limitations of machine learning are being overcome by using Deep Learning techniques.

2. Literature Survey:

The creation of voice assistants began in 1962 at the Seattle World's Fair, when IBM unveiled the shoebox IBM, a device that could recognise spoken digits and return them by lighting lamps labelled adjacent to the digits 0-9. It was capable to understanding a total of 16 words. Most voice assistants are currently designed for mobile phones, such as Google's voice assistance support for Android phones, Apple's Siri, and Amazon's Alexa. These assistants employ language processing to carry out their tasks. Cortana, a voice assistant built by Microsoft and used on PCs, is another option.[1].

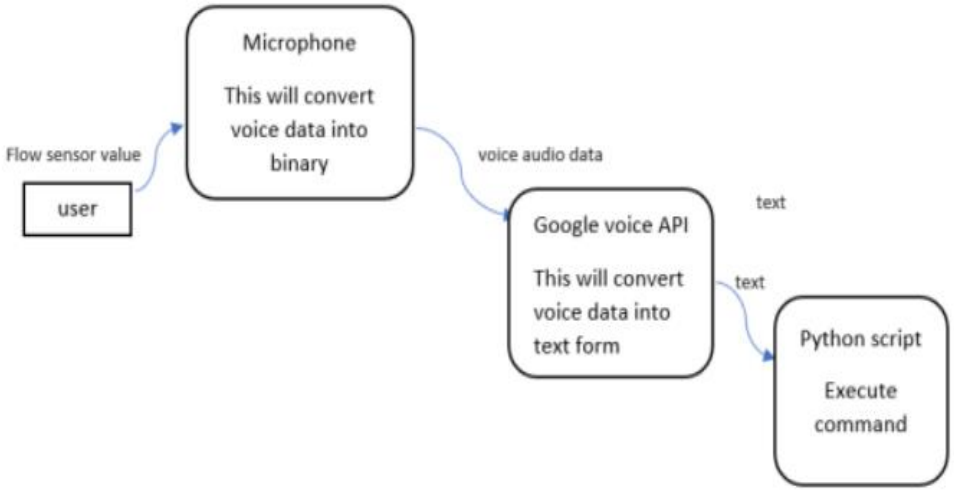
ALICE (Artificial Linguistic Internet Computer Entity) represented input and output with a basic pattern matching algorithm and pattern template. ALICE's recursive procedures are thought to be the system's most important component. Its purpose is to make the input more straightforward. When splitting occurs during the Normalization process, ALICE allows you to merge two replies. Pattern matching algorithms, which are simple and rely on depth first search, are the most significant in ALICE. Srai tags are also included, which can be utilised to reduce the pattern and template size.[2]

Instead of hunting for the short-cut key on the keyboard, the user uses voice command and mouse operation to access the GUI Fig[3] operation. In our system, the user can accomplish the same keyboard command using different mouse and voice operations, with each speech operation mapping to a specific keyboard action and voice operation mapping to a specific keyboard operation. This mouse operation can simply be altered[3].

3. Existing Methodology:

Recognition of speech To translate input speech into text, the suggested system employed the Google API. The speech is sent to Google Cloud for processing, and the system receives the generated text as an output. Backend tasks Python receives the output of speech recognition on the backend and determines whether the command is a system command or a browser command. The output is transmitted back to the Python backend, which provides the user with the desired results. Speech to text Text to speech (TTS) is a relatively new method of converting voice commands into legible text.Not to be confused with VR Systems, which generate speech by combining strings gathered in an extensive database of preinstalled text and have been designed for various purposes, forming full-fledged sentences, clauses, or meaningful phrases using a dialect's graphemes and phonemes. These systems have limitations because they can only determine text based on pre-determined material in databases. TTS systems, on the other hand, "read" strings of characters and produce sentences, clauses, and phrases as a result. ii) Architecture Proposed The system design includes

1. Using the microphone to capture the input as speech patterns.
2. Audio data recognition and text conversion
3. Using specified instructions to compare the input
4. Producing the desired result The data is taken in as speech patterns from the microphone in the first phase. The acquired data is then processed and turned into textual data in the second phase.The obtained stringified data is then processed using Python Script to complete the desired output process. In the last phase, the generated output is provided either as text or as TTS[1] transformed text to speech.



Fig[1]: Data Flow Diagram.

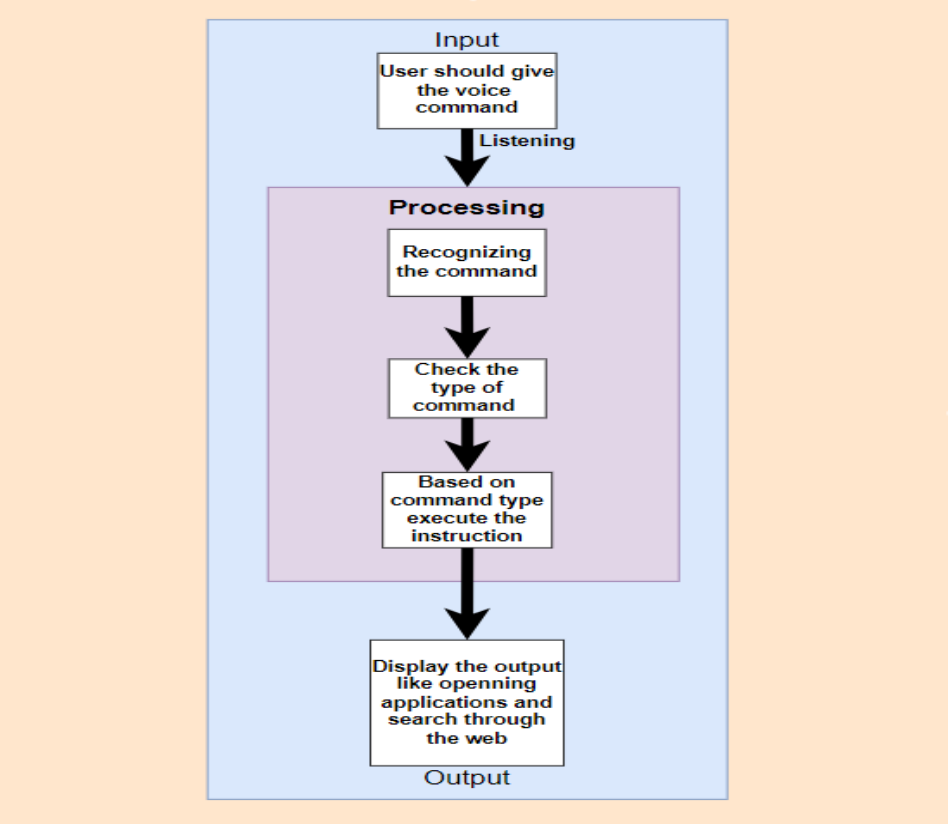
4. Proposed Methodology:

We developed a program that serves the needs of the user, but the user cannot run the code always when he is in need. So we developed GUI(Graphical user interface) Fig[3] by using QWidgets[11]. It's a good tool for developers to design an efficient interface, where the data and the features of the system are represented graphically. So he can directly interact with the interface conveniently. The JSON file is used as a dataset in our project. This includes various kinds of data, and these data are categorized by using Tag or Label, this tells about which kind of data is stored under the name of the tag.

The model which we developed has to be trained. By training, we will get more accurate results and by doing this more and more the Loss function(Expected output-Original output) will get less cost. So we used the Sequential model which is part of Keras. It is a linear pile of layers, where each individual layer has a particular amount of neurons that are used to process the data to get desired results. Keras is an open-source platform that acts as an interface between Python and ANN(Artificial neural network)

[12]. By using this we trained and tested the data stored in the json dataset files. By doing the training of the model along with the model testing, the assistant is ready to answer the user queries by performing some internal operations, such as speech to text conversion it is done by using python inbuilt speech recognition modules and after by using the converted text to identify the words and to categorize them we used lemmatizer which is part of NLP(natural language processing), it then divides into different groups under labels like greetings, googles search, applications, etc.

After identifying the text it is sent to the training model, then we will get to know which intent or tag based on the command or instruction, the model will communicate with the user with voice response by asking for instructions to perform the actions in the system like opening the application or searching for information[5]. It will ask doubts about instructions and after getting clarity then the instruction is based on the intent like application, youtube search Fig[6], google search Fig[4], etc.. it will perform the task related to the intent If it is an application instruction then it will open the application Fig[7][8] and close the application, If it is a youtube search Fig[6] it will go to the youtube and search the required text in the youtube, If it is a google search Fig[4] it will go to the browser and search the required text in the chrome. Every time we use the model it will be getting trained more and more in the time being and it will produce more accurate results.



Fig[2]: System Architecture

4.1.Data set :

To create the data set we used JSON. To produce responses from the system we stored the voice commands in that file. It stores the intents, which consists of tags, utterances along with the related responses[4][5].

4.2.Model creation:

The model consists of speech recognition which is a python module, a speech-to-text converter, and a lemmatizer which can be used for grouping the words, so they can be easily analyzed as a single item.

model = Sequential()

We created this by using the Sequential model as a part of Keras[7]. It operates on a linear layer stack with only one input and output tensor for each layer. Initially, when we create a model, it has no weights. To stack the layers incrementally we have to use the add( ) function.[8]

4.3.Training model: Once the model is created, now its tie to train it by adding layers to themodel[10].

model.add(Dense(128,input_shape=input_shape,activation="relu")) [7].Here we added a layer into the sequential model, the dense job is to make all the possible connections and it is also called a fully connected layer in a neural network. The Relu is an activation function which will return positive numbers and if they are negative then it will result in 0. model.add(Dropout(0.5)).Its job

is to ignore some neurons which are unnecessary which are created in the dence, this will make the model more efficient in making decisions[9].

4.4.Testing model:

We can check the model manually by giving instructions and observing the responses of the system. We observe and check the responses with the commands stored in JSON and calculate the accuracy. It is getting better and better every time it is executed or trained.

5. Results:

5.1. Interface

Fig [3]: GUI

5.2.Input: What is the current time

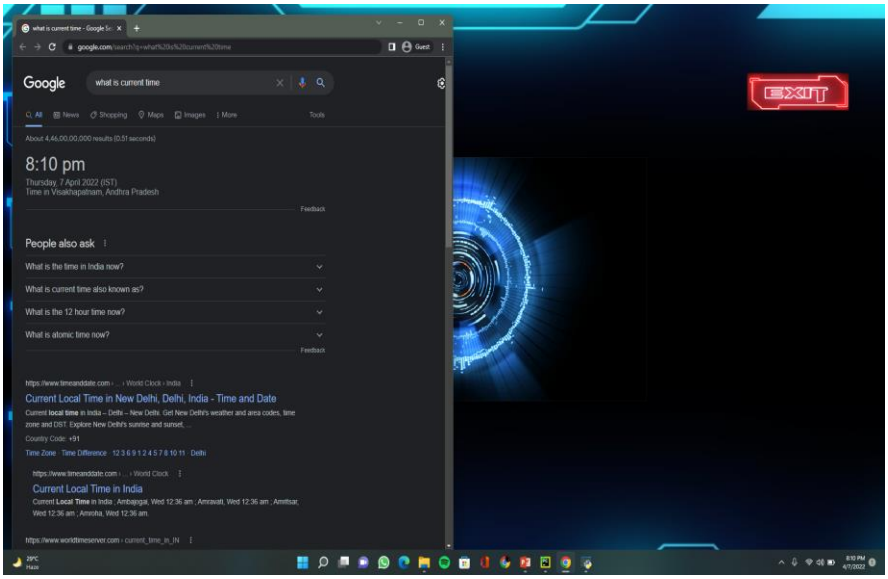


Fig [4]: Google Search for time

5.3.Input: Who is the president of India

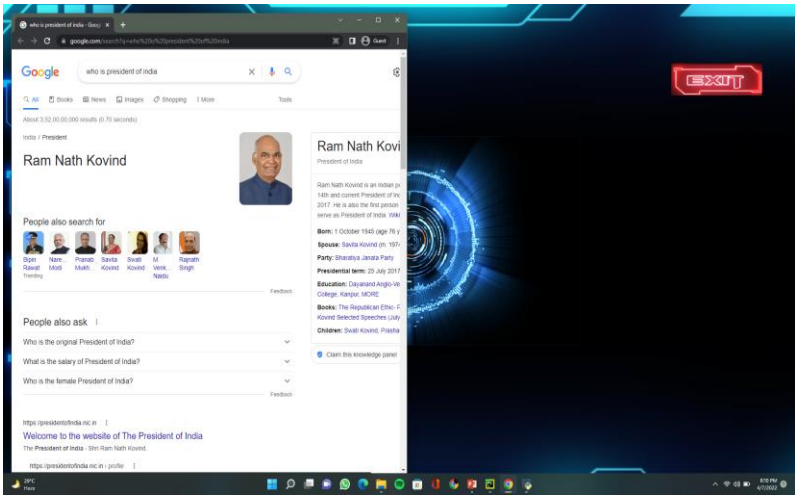


Fig [5]: Google search for information

5.4. Input: Play songs

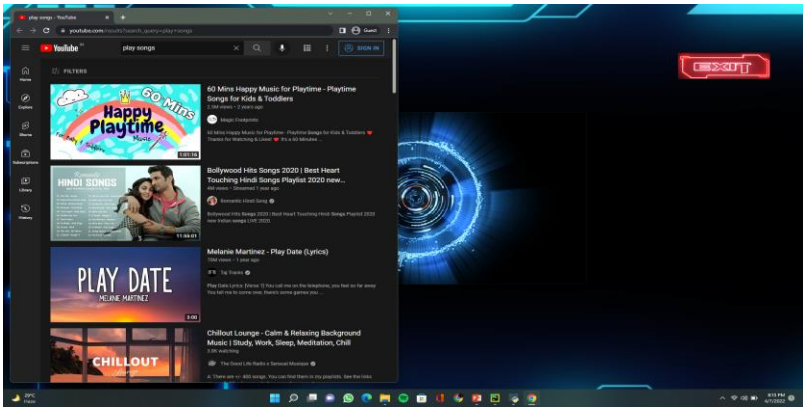


Fig [6]: Youtube Search

5.5. Input: Take a note. Note: my name is assistant.

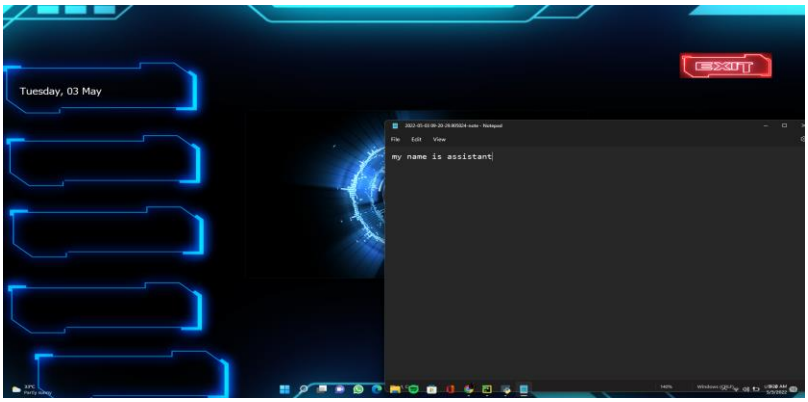


Fig [7]: Opening System Applications

5.6.Input: Open Control Panel:

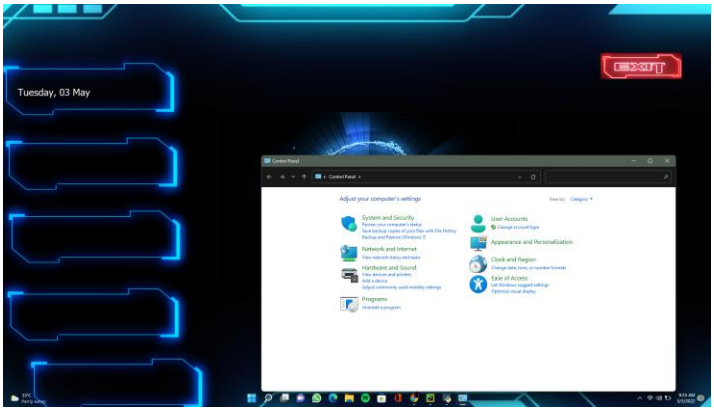


Fig [8]: Opening System App Control panel.

6. Limitations:

There are some limitations with our project they are:

1) At present we have developed an assistant which supports one language, in the near future we can break the barrier by developing a more sophisticated assistant.
2) To use the assistant efficiently we need a strong and durable internet connection, without that we may end up with late responses. To process the voice data and convert it into the text we used python modules, so we need an internet connection. In the future, it is better to overcome this drawback and come up with more efficient solutions.

7. Conclusion:

The project is to build an assistant which is reliable, cost-effective and provides many services to the user. Assistant was used in many applications such as scientific, educational, and commercial. The main agenda of this project is to serve people with physical disabilities. If a person is facing difficulty in communicating with the system it is not proper communication. To overcome this kind of issue we developed an assistant which helps them to communicate with the system efficiently.This assistant responses from a finite set of predefined responses or pre-existing Information and training. Assistants are a big step forward in enhancing human computer interactions.

8. References:

[1].https://www.ijrte.org/wp-content/uploads/papers/v9i2/A2753059120.pdf
[2].https://media.neliti.com/media/publications/263312-an-intelligent-behaviour-shown-by-chatbo-7020467d.pdf
[3].Apte, T. V., Ghosalkar, S., pandey, S., & Padhra, S. (2014). Android app for blind using speech technology. International Journal of Research in Computer and Communication Technology (IJRCCT), #(3), 391-394.
[4].https://opensource.adobe.com/Spry/samples/data_region/JSONDataSetSample.html

[5].https://support.etlworks.com/hc/en-us/articles/360014078293-JSON-dataset-Format#:~:text=The%20JSON%20dataset%20is%20a,REST%20APIs%20in%20Etlworks%20Integrator
[6].Anwani, R., Santuramani, U., Raina, D., & RL, P. Vmail: voice Based Email Application. International Journal of Computer Science and Information Technologies, Vol. 6(3), 2015
[7]. https://keras.io/guides/sequential_model/
[8]. https://keras.io/guides/making_new_layers_and_models_via_subclassing/
[9]. https://keras.io/guides/training_with_built_in_methods/
[10]. https://analyticsindiamag.com/a-tutorial-on-sequential-machine-learning/
[11]. https://doc.qt.io/qtforpython-5/PySide2/QtWidgets/QWidget.html
[12].https://www.sciencedirect.com/topics/earth-and-planetary-sciences/artificial-neural-network