

Voice Assistant

Principle of Programming Language

CS300

Evaluation Report

Presented by –

Sudhir Sharma 12041500

Suresh Bairwa 12041530

Akash Deep 12040060

Abstract

In this modern era, day-to-day life became smarter and interlinked with technology. We already know some voice assistants like google and Siri. etc. Now in our voice assistance system, it can act as a basic medical prescriber, daily schedule reminder, note writer, calculator, and search tool. This project works on voice input and gives output through voice and displays the text on the screen. The main agenda of our voice assistance makes people smart and give instant and computed results. The voice assistance takes the voice input through our microphone (Bluetooth and wired microphone) and it converts our voice into computer understandable language gives the required solutions and answers which are asked by the user. This assistance connects with the world wide web to provide results that the user has questioned. Natural Language Processing algorithm helps computer machines to engage in communication using natural human language in many forms.

I. Introduction

Today the development of artificial intelligence (AI) systems that can organize a natural human-machine interaction (through voice, communication, gestures, facial expressions, etc.) are gaining in popularity. One of the most studied and popular was the direction of interaction, based on the understanding of the machine by the machine of the natural human language. It is no longer a human who learns to communicate with a machine, but a machine learns to communicate with a human, exploring his actions, habits, behaviour and trying to become his personalized assistant.

II. Proposed Plan Of Work

The work started with analysing the audio commands given by the user through the microphone. This can be anything like getting any information, operating a computer's internal files, etc. This is an empirical qualitative study, based on reading above mentioned literature and testing their examples. Tests are made by programming according to books and online resources, with the explicit goal to find best practices and a more advanced understanding of Voice Assistant.

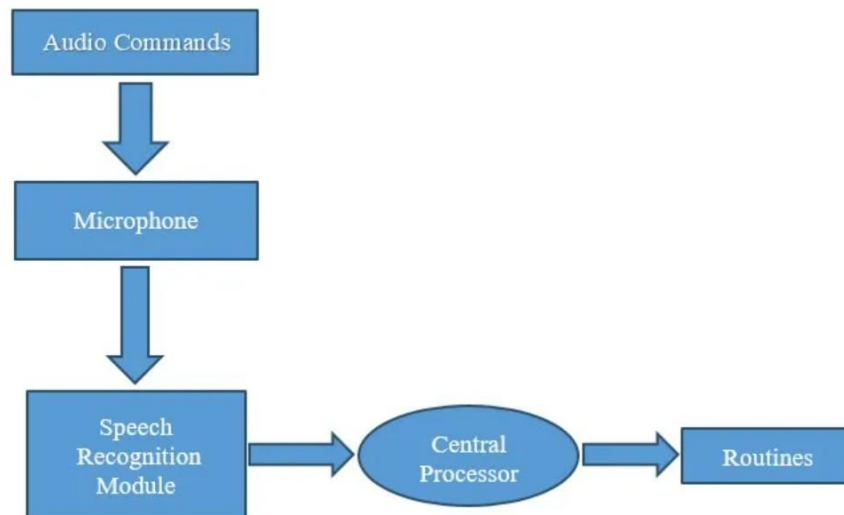


Fig.2. Basic Workflow

Fig.2 shows the workflow of the basic process of the voice assistant. Speech recognition is used to convert the speech input to text. This text is then fed to the central processor which determines the nature of the command and calls the relevant script for execution.

But, the complexities don't stop there. Even with hundreds of hours of input, other factors can play a huge role in whether or not the software can understand you. Background noise can easily throw a speech recognition device off track. This is because it does not inherently have the ability to distinguish the ambient sounds it "hears" of a dog barking or a helicopter flying overhead, from your voice. Engineers have to program that ability into the device; they conduct data collection of these ambient sounds and "tell" the device to filter them out. Another factor is the way humans naturally shift the pitch of their voice to accommodate for noisy environments; speech recognition systems can be sensitive to these pitch changes.

III. Methodology of Virtual Assistant Using Python

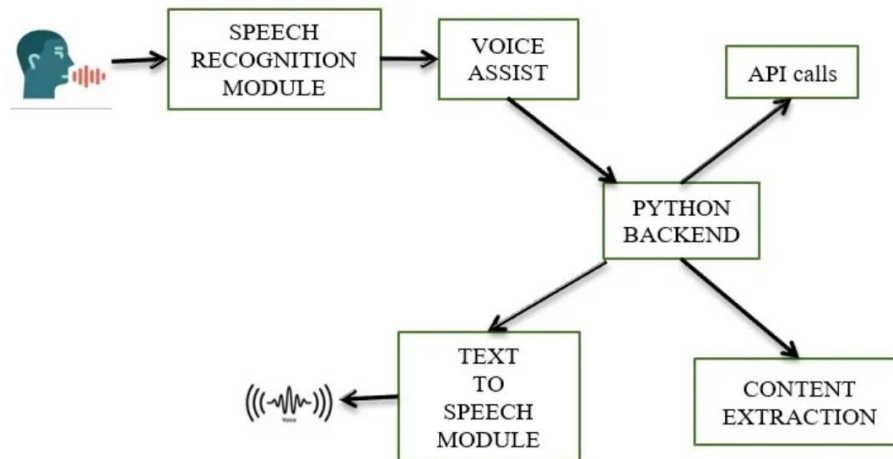


Fig 3 Detailed workflow

Fig 3 Detailed workflow

Speech Recognition module

The system uses Google's online speech recognition system for converting speech input to text. The speech input Users can obtain texts from the special corpora organized on the computer network server at the information centre from the microphone is temporarily stored in the system which is then sent to Google cloud for speech recognition. The equivalent text is then received and fed to the central processor.

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.

API calls

API stands for Application Programming Interface. An API is a software intermediary that allows two applications to talk to each other. In other words, an API is a messenger that delivers your request to the provider that you're requesting it from and then delivers the response back to you.

Content Extraction

Context extraction (CE) is the task of automatically extracting structured information from unstructured and/or semi-structured machine-readable documents. In most cases, this activity concerns processing human language texts using natural language processing (NLP). Recent activities in multimedia document processing

like automatic annotation and content extraction out of images/audio/video could be seen as context extraction TEST RESULTS.

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

IV. Conclusion

In this paper “Virtual Assistant Using Python” we discussed the design and implementation of Digital Assistance. 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 easy to add additional features without disturbing current system functionalities.

It not only works on human commands but also give responses to the user based on the query being asked or the words spoken by the user such as opening tasks and operations. It is greeting the user the way the user feels more comfortable and feels free to interact with the voice assistant. The application should also eliminate any kind of unnecessary manual work required in the user life of performing every task. The entire system works on the verbal input rather than the next one.