BUILDING AN INTELLIGENT DESKTOP VOICE ASSISTANT USING NLP AND AI

DONE BY

K.Abhinay

ABSTRACT

The rapid evolution of artificial intelligence has made possible the development in speech recognition technology which is now actively penetrating almost every area of human life. Voice assistant is an important

achievement in this field. A voice assistant is digital assistant that uses speech recognition, speech synthesis, and Natural Language Processing to provide user a required information or to perform any task like getting weather updates, getting traffic updates, creating remainder, sending mails, making calls and many other services just on a simple voice command.

The system first uses speech recognition for converting the voice command into text for processing. Voice assistant relies on Natural Language Processing to resolve the barriers of understanding. After processing, voice assistant retrieves information related to the question using the knowledge base or various Application Programming Interfaces, if the command is task-based then the system calls are used. Finally, speech synthesis converts the output or result in speech format from the text. This system provides user hands-free access to various services.

The result of a case study shows that the proposed voice assistant system can effectively utilize the user's command, Application Programming Interfaces or system calls depending on user requirement to provide the services.

KEYWORDS: Artificial Intelligence, Voice assistant, Speech recognition, Speech synthesis, Natural Language Processing

CONTENTS

ABBREVIATIONS

Abbreviations Meaning

AI Artificial Intelligence

NLP Natural Language Processing

STT Speech-to-Text

TTS Text-to-Speech

DSP Digital Signal Processing

API Application Programming Interface

GUI Graphical User Interface

URL Uniform Resource Locator

IDE Integrated Development Environment

LIST OF FIGURES

Figure

CHAPTER 1

INTRODUCTION

1.1 Voice Assistant

Voice assistant is a task-oriented programming application or software that recognizes human speech and carries out commands pronounced by a user. It is powered by AI and bases its performance on cloud storage with millions of words and phrases in it. The rapid evolution of AI and machine learning made possible the development of speech recognition technology, which is how actively penetrating every area of our lives.

And there's nothing to wonder about; for such social-dependent creatures as humans, speaking is a lot more natural activity than writing or of course,typing. An average human can type about 40 words in one minute, but pronounce 150. This contrast, alongside with many other benefits, vividly demonstrates why voice technology should be taken seriously. Voice assistants are not to be confused with virtual assistants, which are people who work remotely and can therefore handle all kinds of tasks. Rather, voice assistants are technology based. As voice assistants become more robust, their utility in both the personal and business realms will grow as well. The most common voice assistants in today's world are Siri by Apple, Google Assistant by Google, Alexa by Amazon, Samsung's Bixby, Microsoft's Cortana.

1.2 History of Voice Assistants

Voice assistants have a very long history that actually goes back over 100 years, which might seem surprising as apps such as Siri have only been released within the past ten years. The very first voice activated product was released in 1922 as Radio Rex. This toy was very simple, wherein a toy dog would stay inside a dog house until the user exclaimed its name, "Rex" at which point it would jump out of the house. IBM

began their long history of voice assistants in 1962 at the World's Fair in Seattle when IBM Shoebox was announced This device was able to recognize digits 0-9 and six simple commands such as, "plus, minus" so the device could be used as a simple calculator.

Darpa then funded five years of speech recognition R&D in 1971, known as the Speech Understanding Research (SUR) Program. One of the biggest innovations to come out of this was Carnegie Mellon's Harpy, which was capable of understanding over 1,011 words.

The next decade led to amazing progress and research in the speech recognition field, leading most devices from understanding a few hundred words to understanding thousands all with the help of AI technology. In 1994, Simon by IBM was the first smart voice assistant and really, the first smartphone in history.

The next decade led to amazing progress and research in the speech recognition field, leading most devices from understanding a few hundred words to understanding thousands all with the help of AI technology. In 1994, Simon by IBM was the first smart voice assistant and really, the first smartphone in history. In 2008, when Android was first released, Google had slowly started rolling out voice search for its Google mobile apps on various platforms, with a dedicated Google Voice Search Application being released in 2011. This led to more and more advanced features, eventually leading to Google now and Google Voice Assistant.

Then, this was followed by Siri in 2010.Developed by SRI International with speech recognition provided by Nuance Communications, the original app was released in 2010 on the iOS App Store and was acquired two months later by Apple. Then, with the release of the iPhone 4s, Siri was officially released as an integrated voice assistant within iOS. Since then, Siri has made its way to every Apple device available and has linked all the devices together in a single ecosystem

Shortly after Siri was first developed, IBM Watson is announced publicly in 2011. Watson was named after the founder of IBM, and was originally conceived in 2006 to beat humans at a game of Jeopardy. Now, Watson is one of the most intelligent, naturally speaking computer systems available. Amazon

Alexa is then announced in 2015, helping with more accurate speech recognition. With Alexa Echo, the line of smart devices is announced to bring smart integration.

1.3 Overview of Proposed System

The Intelligent Desktop Voice Assistant is developed and designed to assist the user with the basic tasks. The proposed system take voice as input key from the user, this is done Speech recognition**.** The input is then processed by Natural Language Processing followed by Speech Synthesis for providing the required output in the form of voice, thereby providing hands free access to user

This conversational voice assistant is combination of both a task-oriented and knowledge-oriented workflow to carry out almost every task that user throw at it. A task-oriented approach workflow includes task like writing and sending the E-mail, while knowledge-oriented workflow includes answering 'What is my current location?' or solving the mathematical calculations.

1.4 Motivation

Nowadays, user mainly uses voice assistant which comes in built in the specific device. They are bound to use the services which are available in particular voice assistant but not which they prefer. Also, switching among different voice assistant ultimately results in switching manufacturer of that particular device. This process of switching voice assistant faces two main challenge, one is adapting to change in system environment and second is high cost requirement**.**

1.5 Organization of Report

Chapter one contains the introduction of the proposed system which comprises information regarding the voice assistants, their history and overview. This chapter also describes the motivation and scope for the project.

Chapter two includes the literature review which refers to the study that has been carried out on different voice assistant systems over the previous years that have some relevance with existing system and also various technologies which are used for proposed system. This chapter also specifies the aim and objectives.

Chapter three explains the tools and technologies which are used for developing voice assistant. It also explains about the APIs.

Chapter four explains proposed approach and system architecture which includes the detail of how the voice assistant has been developed with different components.

Chapter five discusses the implementation details of proposed system, this chapter explains the coding part developed for performing various functions.

Chapter six discusses the results that were generated from the proposed system of voice assistant and showcases every necessary output needed to describe the proposed system precisely.

CHAPTER 2

LITERATURE REVIEW

A literature review discusses published information in a particular subject area, and sometimes information in a particular subject area within a certain time period. It can be just a simple summary of the sources, but it usually has an organizational pattern and combines both summary and synthesis. A summary is a recap of the important information of the source, but a synthesis is a re -organization, or a reshuffling of that information

2.1 Related Work:

2.1.1 Voice Assistant for Home Automation:

Prerna Wadikar, Nidhi Sargar, Rahool Rangnekar, Prof. Pankaj Kunekar (2020) “Home Automation using Voice Commands in the Hindi Language”. The proposed project of Home Automation in Hindi language. Voice commands was to implement the dedicated hardware i.e. Arduino Uno and using voice recognition module that makes the system more cost-efficient and robust. The system can work on various connected devices like light, fan, AC, etc. This system allows users to make decisions and to regulate the home appliances with the help of voice assistants.

2.1.2 Voice Assistant for Blind People:

Steve Joseph, Chetan Jha, Dipesh Jain, Saurabh Gavali, Mapythnish Salvi (2020) proposed “Voice based E-Mail for the Blind”. They designed the system that was helpful for sending emails for the blind people without the need of visual interaction with the screen. Speech-to-Text based Life Log System for Smartphones. The

technique used in it was Microphone of Smartphone, STT (Speech-to-Text)

Nishank Tembhurne, Sumedh Vaidya, Afrin Shiekh, Prof. Swapnil Dravyakar (2019) developed a customized application “Voice Assistant for Visually Impaired People”. This application is used to help the visually impaired to access most important features of the phone using text to speech and speech to text. The system had custom messaging feature, call log feature, notes making feature, OCR feature, web browsing feature, navigation feature in it. The custom app having these features made it possible for visually impaired users to do their basic things using electronic device without any other help.

2.1.3 Voice Assistant for Android and Linux:

Shen Hui, Song Qunying and Andreas Nilsson (2012) proposed an“Intelligent Voice Assistant” application for androids. This project focused on the Android development over the voice control, relevant APIs and mobile device references ranging from Speech-To-Text, Text-To-Speech technology, Bluetooth headset support and camera; advanced techniques of Cloud computing and Multi-threading. This proposed system was developed by various technologies like Java, Android development, MySQL and Network connection technologies

2.2 Artificial Intelligence:

In computer science, the term artificial intelligence (AI) refers to any human-like intelligence exhibited by a computer, robot, or other machine. In popular usage, artificial intelligence refers to the ability of a computer or machine to mimic the capabilities of the human mind; learning from examples andexperience, recognizing objects, understanding and responding to language, making decisions, solving problem and combining these and other capabilities to perform functions a human might perform.

Deep learning techniques enable this automatic learning through the absorption of huge amounts of unstructured data such as text, images, or video. After decades of being relegated to science fiction, today, AI is part of our everyday lives. The surge in AI development is made possible by the sudden availability of large amounts of data and the corresponding development and wide availability of computer systems that can process all that data faster and more accurately than humans.

Artificial Intelligence is divided into two different categories: weak and strong. Weak artificial intelligence also called as Narrow AI, embodies a system designed to carry out one particular job. Weak AI systems include video games such as the chess example from above and personal assistants such as Amazon's Alexa and Apple's Siri. Strong artificial intelligence or Wide AI systems are systems that carry on the tasks considered to be human-like.

2.3 Speech Recognition:

Speech recognition, also known as automatic speech recognition, computer speech recognition, or speech-to-text, is a capability which enables a program to process human speech into a written format. While it's commonly confused with voice recognition, speech recognition focuses on the translation of speech from a verbal format to a text one whereas voice recognition just seeks to identify an individual user's voice.

They integrate grammar, syntax, structure, and composition of audio and voice signals to understand and process human speech. The vagaries of human speech have made development challenging. It's considered to be one of the most complex areas of computer science involving linguistics, mathematics and statistics.

Natural Language Processing:

While NLP isn't necessarily a specific algorithm used in speech recognition,it is the area of artificial intelligence which focuses on the interaction between humans and machines through language through speech and text. Many desktop and mobile devices incorporate speech recognition into their systems to conduct voice search e.g. Siri or provide more accessibility around texting. Natural language processing layers increasingly powerful and complex systems to handle tasks that require higher levels of understanding.

. Hidden Markov Models:

Hidden Markov Models build on the Markov chain model, which stipulates that the probability of a given state hinges on the current state, not its prior states. While a Markov chain model is useful for observable events, such as text inputs,

Hidden Markov Models allow us to incorporate hidden events, such as part-of-speech tags, into a probabilistic model.

. N-grams:

This is the simplest type of language model, which assigns probabilities to sentences or phrases. An N-gram is sequence of N-words. For example, “order the pizza” is a trigram or 3-gram and “please order the pizza” is a 4-gram. Grammar and the probability of certain word sequences are used to improve recognition and accuracy**.**

Neural networks:

Primarily leveraged for deep learning algorithms, neural networks process training data by mimicking the interconnectivity of the human brain through layers of nodes. Each node is made up of inputs, weights, a bias (or threshold) and an output. If that output value exceeds a given threshold, it “fires” or activates the node, passing

Speaker Diarization:

Speaker Diarization algorithms identify and segment speech by speaker identity. This helps programs better distinguish individuals in a conversation and is frequently applied at call centers distinguishing customers and sales agents**.**

2.4 Speech Synthesis:

Speech synthesis or text-to-speech abbreviated as TTS, is defined as the artificial production of human voices. The main use is the ability to translate a text into spoken speech automatically. Unlike speech recognition systems that use phonemes in the first place to cut out sentences, TTS will be based on what are known as graphemes: the letters and groups of letters that transcribe a phoneme. This means that the basic resource is not the sound, but the text.

The best way to gain acceptance for voice assistants was to offer new features that promote their use, but also to improve the user experience as much as possible by humanising the technology. These synthesised voices then made it possible to give an identity to the various assistants, making it possible to differentiate them, but also to consider them as entities in their own right.

2.5 Natural Language Processing:

Natural language processing shortened as NLP refers to the branch of computer science and more specifically, the branch of AI concerned with giving computers the ability to understand text and spoken words in much the same way human beings can. NLP combines computational linguistics rule-based modeling of human language with statistical, machine learning, and deep learning models.

Input and Initial Processing

Taking in speech or text and breaking it up into smaller pieces for processing. For speech, this step is called phonetic analysis, and consists of breaking down the speech into individual sounds, called phonemes. For text input, this can include optical character recognition (OCR) and tokenization. OCR is used to recognize the individual characters in text if it's coming in as an image rather than as words made of characters. Tokenization refers to breaking down a continuous text into individual tokens, often words.

Syntactic Analysis:

Trying to understand the structure of sentences by looking at how the words work together. This step is like diagramming a sentence, where you identify the role each word is playing in the sentence.

Semantic Interpretation:

Working out the meaning of a sentence by combining the meaning of individual words with their syntactic roles in the sentence.

CHAPTER 3

TOOLS AND TECHNOLOGIES

This chapter describes various technologies those are being used in the development of the proposed system. The function and modules are explained along with features and components.

3.1 Python Programming Language

Python is a general purpose, dynamic, high-level, and interpreted programming language. It supports object oriented programming approach as well as functional programming approach. Python's syntax and dynamic typing with its interpreted nature make it an ideal language for scripting and rapid application development. Python supports multiple programming pattern, including object-oriented, imperative, and functional or procedural programming styles. Python is not intended to work in a particular area, such as web programming. That is why it is known as general purpose programming language.

3.1.1 Applications of Python:

There are so many applications of Python some of them are as follows:

1. Web Development

Web framework like Django and Flask are based on Python. They help us to write server side code which helps us to manage database, write backend programming logic, mapping urls etc. Python web frameworks are known for their security, scalability and flexibility.

2. Machine Learning

There are many machine learning applications written in Python. Machine learning is a way to write a logic so that a machine can learn and solve a particular problem on its own. For example, voice recognition technology in Apple's Siri, Google Assistant.

Game Development

Python comes loaded with many useful extensions that come in handy for the development of interactive games. For instance, libraries like PySoy and PyGame are two Python-based libraries used widely for game development.

4. Scientific and Numeric Applications

Python has become a crucial tool in scientific and numeric computing. In fact, Python provides the skeleton for applications that deal with computation and scientific data processing. Libraries like SciPy, Pandas, Natural Language Toolkit etc. are most useful for scientific and numeric applications.

3.1.2 Features of Python

1. Easy to Learn and Use

Python is easy to learn as compared to other programming languages. Its syntax is

straightforward and much the same as the English language. There is no use of the

semicolon or curly-bracket, the indentation defines the code block.

2. Expressive Language

Python can perform complex tasks using a few lines of code. A simple example is hello world program it take only one line to execute, print("hello world").

3. Cross-platform Language

Python can run equally on different platforms such as Windows, Linux, UNIX and Mac etc. Python is a portable language. It enables programmers to develop the software for several competing platforms by writing a program only once.

4. Open Source

Python is freely available for everyone. It is freely available on it official website www.python.org

3.2 Visual Studio Code IDE

Visual Studio Code is a source-code editor that can be used with a variety of programming languages, including Java, JavaScript, Go, Node.js, Python and C++. It is based on the Electron framework, which is used to develop Node.js Web 1. applications that run on the Blink layout engine. Visual Studio Code employs the same editor component used in Azure DevOps. Instead of a project system, it allows users to open one or more directories, which can then be saved in workspaces for future reuse. This allows it to operate as a language-agnostic code editor for any language.

3.3 Python Modules

1. Pyttsx3

pyttsx3 is a text-to-speech conversion library in Python. Unlike alternative libraries, it works offline and is compatible with both Python 2 and 3. An application invokes the pyttsx3.init() factory function to get a reference to a pyttsx3. Engine instance. it is a very easy to use tool which converts the entered text into speech. The pyttsx3 module supports two voices first is female and the second is male which is provided by “sapi5” for windows. It supports three TTS engines:

. sapi5: SAPI5 on Windows

. nsss: NSSpeechSynthesizer on Mac OS X

. espeak: eSpeak on every other platform

Speech_recognition

Speech recognition takes input in form of voice and then convert it to text. speech-recognition is simplest of all the libraries having same purpose. It is capable of recognizing speech from audio files, as well as live from a microphone. Then the speech to text translation is done with the help of Google Speech Recognition. This requires an active internet connection to work. There are several other offline Recognition systems such as PocketSphinx, but have a very rigorous installation process that requires several dependencies. Google Speech Recognition is one of the easiest to use.

Datetime

In Python, date and time are not a data type of its own, but a module named datetime can be imported to work with the date as well as time.

Datetime module comes built into Python. This module supplies classes to work with date and time.The datetime classes are categorize into 6 main classes:

. date: An idealized naive date, assuming the current Gregorian calendar always was, and always will be, in effect. Its attributes are year, month and day.

.time: An idealized time, independent of any particular day, assuming that every day has exactly 24*60*60 seconds. Its attributes are hour, minute, second,microsecond.

. datetime: Its a combination of date and time along with the attributes year, month, day, hour, minute, second, microsecond, and tzinfo.

. timedelta: A duration expressing the difference between two date, time, or datetime instances to microsecond resolution.

. timezone: A class that implements the tzinfo abstract base class as a fixed offset.

OS:

The OS module in Python provides functions for interacting with the operating system. OS comes under Python's standard utility modules. This module provides a portable way of using operating system-dependent functionality. The 'os' and 'os.path' modules include many functions to interact with the file system.

.os.name()

This function provides the name of the operating system module that it imports.

.os.mkdir()

The os.mkdir() function is used to create new directory

. os.getcwd()

It returns the current working directory(CWD) of the file.

.os.chdir()

The os module provides the chdir() function to change the current working directory.

OpenWeatherMap

OpenWeatherMap provides a range of weather-related products in a variable combination of depth and steps of measurement to millions of clients globally. The product range includes current, historical and forecasted weather data with the granularity as high as 1 minute. The length of the nowcast reaches 2 hours, short-term forecast reaches 16 days and long-term forecast can reach up to 1 year length. Historical weather data goes over 40 years deep Learning .

Open Weather also operates under the terms of license providing free access to the APIs that include current weather, a minutely forecast for 1 hour, hourly forecast for 48 days, 3-hour forecast for 5 days, daily forecast for 7 days, short-term history, weather maps, alerts, geocoding, air quality weather triggers and weather widgets. The projects with a higher demand of loading, may obtain an extended service on the basis of paid subscription.

CHAPTER 4

PROPOSED APPROACH AND SYSTEM ARCHITECTURE

This chapter represents the architecture of proposed system. It explains the various components and models involve in development of voice assistant in detail. It also describes the approach followed by proposed system.

4.1 Proposed Approach

The proposed system uses the combination of both Task-oriented approach and Knowledge-oriented approach to perform almost every task that user wants to get done.

4.1.1 Task-oriented approach

A task-oriented approach use goals to tasks to achieve what the user needs. This approach often integrates itself with other apps to help complete tasks. For example, if user asks a voice assistant to set an alarm for 3PM, it understand this to be a task request and communicate with the default Clock application to open and set an alarm for 3PM. This approach does not require an extensive online APIs, as it is mainly using the system calls and already existing skills of other installed applications.

4.1.2 Knowledge-oriented approach

A knowledge-oriented approach is the use of analytical data to help users with their tasks. This approach focuses on use of online APIs and already recorded knowledge to help complete tasks. An example of this approach is anytime a user asks for an internet search, it uses the online APIs available to return relevant results and recommend the highest search result. If user searches up a trivia question, this process need knowledge-oriented approach as it is searches for data instead of working with other apps to complete tasks.

4.1.3 Proposed System and Existing System

• One of the main benefit of proposed voice assistant is that the user can personalise the functions of system depending upon the need. User can also control the system'sway of responding back along with the voice that system uses, providing the more human-like conversational environment.

Architecture of Proposed System

The overall architecture of proposed system consists of following phases:

1. Speech –to –text

Taking input from the user in form of voice and converting the speech into text to be processed by the assistant. Voice assistant uses the microphone as source for taking the voice command as the input. This phase then uses speech recognition system for converting the speech to text for further processing.

2. Command processing and execution

Next is processing the converted text to get results. The text contains one or two keywords that determine what query is to be executed. If the keyword doesn't match any of the queries in the code, then the assistant asks user to speak again. The proposed system involves the task-oriented as well as knowledge-oriented approach. Thus can easily solve questions using APIs and can perform tasks like opening desktop application using system calls.

3. Text-to-Speech

The output which is in form of text is then converted to speech to give final result. This phase uses the speech synthesis system for converting the text to speech. The proposed system not only speaks out result but also displays it on screen.

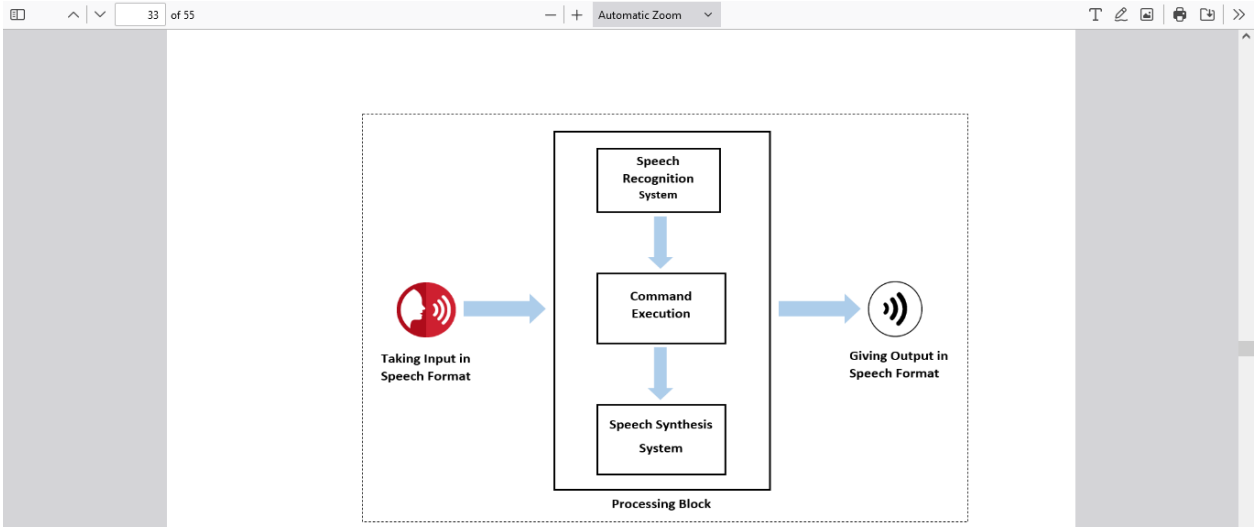


Fig.4.1 Architecture of proposed system

4.3 Architecture of Speech Recognition System

Speech recognition, also known as automatic speech recognition or speech-to-text, is ability of machine which enables a program to process human speech into a written format by identifying the words spoken.

Fig.4.2 Speech-To-Text system

The recognition system has three separate models:

• Acoustic Modeling / Phoneme Detection

• Pronunciation Modeling

• Language Modeling

These three models are trained separately, but are then composed into one gigantic search graph. Essentially, speech recognition is taking an audio waveform, pushing it through this search graph, and letting it find the path of least resistance—that is, finding the word sequence that has the maximum likelihood. These three models create a huge search graph, through which waveforms can be pushed to create near instantaneous text output.

4.3.1 Acoustic Model:

An acoustic model is use to represent the relationship between an audio signaland the phonemes or other linguistic units that make up speech. User speaks to the computer using a microphone. The microphone converts the sound signals into electrical signals. It takes a waveform, chunks it into small time-segments, implements a frequency analysis, and outputs a probability distribution over all the triphone-states for that particular input. The phonemes are the sound units that distinguish one word from the other. The waveform frequency vector, matched with a probability distribution, thus identifies which phonemes are more likely than others to be contained in the audio sample, thereby delivering a sequence of phonemes that exist in that input over time

4.3.2 Pronunciation Model

A word pronunciation is a possible phoneme-like sequence that can appear in a real utterance and represents a possible acoustic pronunciation of the word. The common dictionary will be created using the words with the accurate phonemes connections called canonical pronunciation. The pronunciation model can be used to map the phoneme-like units to the standard pronunciation that can be found in the common dictionary. These mapping the phoneme units to the correct pronunciation of the word are tougher than the other processes.

4.4 Architecture of Speech Synthesis System

Speech synthesis is the artificial production of human speech or text-to-speech system converts normal language text into speech. Here, the input is text and output is speech.

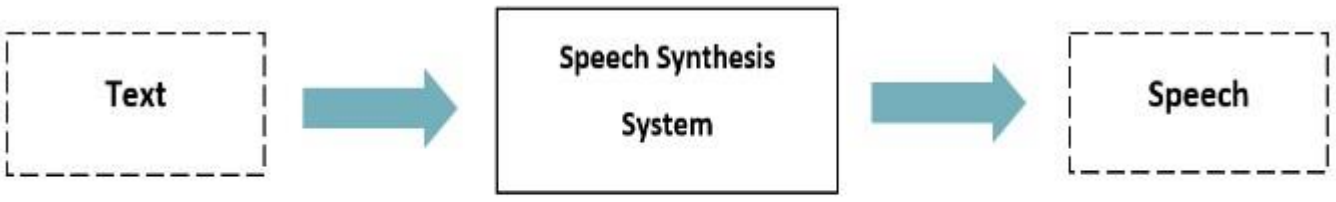


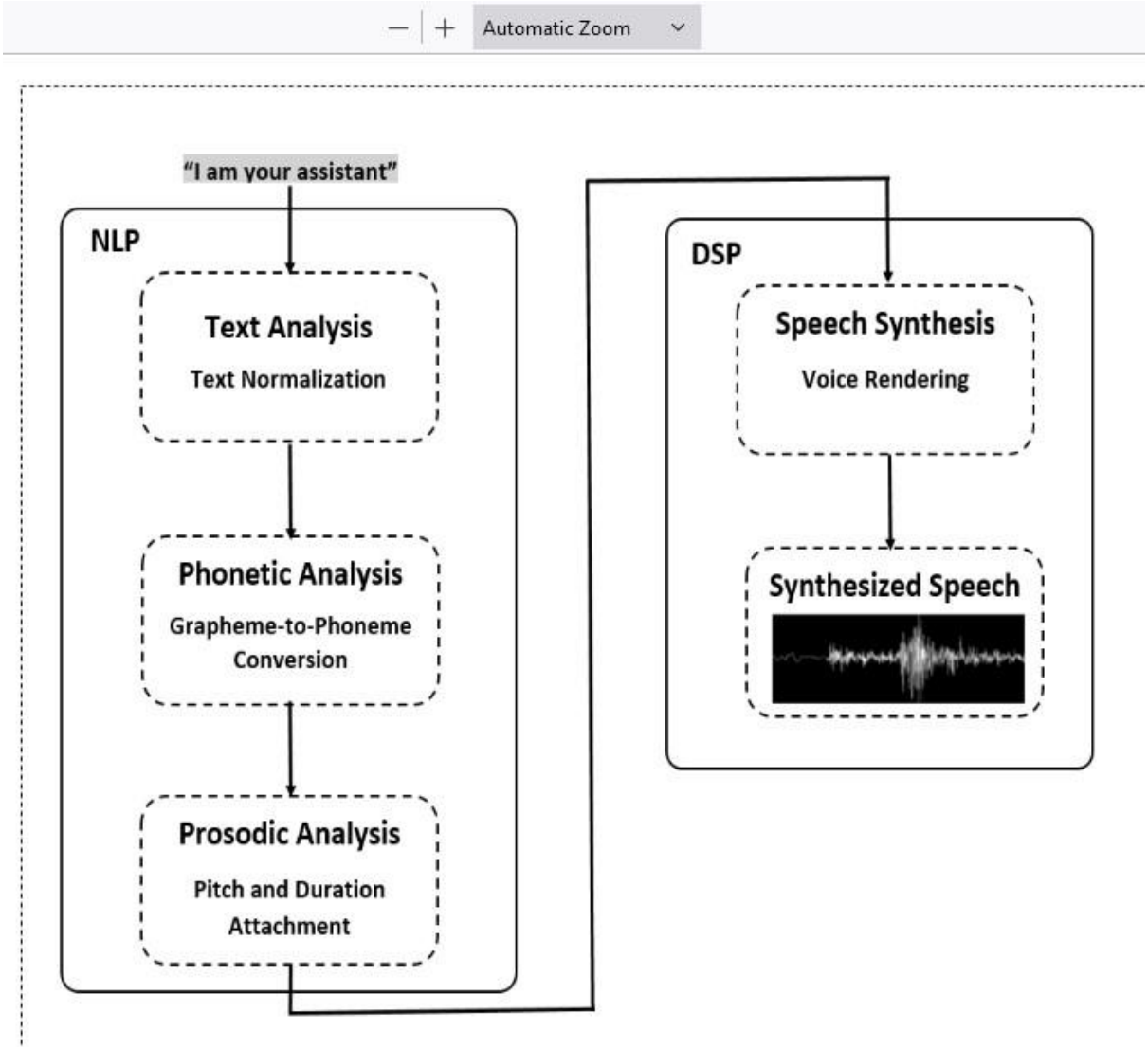
Fig.4.4 Text-to-Speech system

A text-to-speech system is composed of two parts:

- Natural Language Processing Module
- Digital Signal Processing Module

NLP module has two major tasks, firstly, it converts raw text containing symbols, such as numbers and abbreviations, into the equivalent of written words. This process is often called text normalization or text analysis. This process then assigns phonetic.

transcriptions to each word, and divides and marks the text into prosodic units, like phrases, clauses, and sentences. The process of assigning phonetic transcriptions to words is called text-to-phoneme or grapheme-to-phoneme conversion. Phonetic transcriptions and prosodic information together make up the symbolic linguistic representation that gives output of NLP module. Digital Signal Processing module often referred to as the synthesizer, then converts the symbolic linguistic representation into sound. Text-to-speech synthesis takes place in several steps.



4.4.1 Natural Language Processing Module

It produces a phonetic transcription of the text read, together with prosody. The major operations of the NLP module are as follows:

• Text Analysis or Text Normalisation

All written languages have special constructs that require a conversion of the written form or orthographic form into the spoken form. First the text gets segmented into tokens. The token-to-word conversion creates the orthographic form of the token. For the token “Mr” gives the orthographic form “Mister” by expansion, the token “12” gives the orthographic form “twelve”, for “$200” it says “200 hundred dollars” and “1997” transforms to “nineteen ninety-seven”.

• Phonetic Analysis

Phonetic analysis or grapheme-to-phoneme conversion or text-to-phoneme conversion is the process of generating pronunciation for words based on their written form. The spelling of a word is called a grapheme and the phonetic form is called phonemes. Speech synthesis systems uses two basic approaches to determine the pronunciation of a word based on its spelling. The simplest approach to text-to- phoneme conversion is the dictionary-based approach, where a large dictionary containing all the words of a language and their correct pronunciations is stored in program. The other approach is rule-based, in which pronunciation rules applied to words determines their pronunciations based on their spellings. This is similar to the synthetic phonics, approach to learning reading. Proposed system uses combination of these approaches for getting more accurate result and handling exception.

• Prosodic Analysis

Prosody is the set of features of speech output that includes the pitch also called intonation, the timing, the pausing, the speaking rate, the emphasis on words and many other features. Producing human-like prosody is important for making speech sound natural and for correctly conveying the meaning of spoken language. This can be achieved through analysis of the document structure, sentence syntax, and other information that can be inferred from the text input.

4.4.2 Digital Signal Processing Module

It transforms the symbolic information it receives from NLP into audible and intelligible speech.

• Waveform Generation

Its task is to select the appropriate sequence of units from the inventory; modify the pitch, duration of each unit; and concatenate these modified units to produce the desired speech waveform. Thus, the final output of this step is speech

CHAPTER 5

IMPLEMENTATION

It represents the development stages of proposed system. This chapter also describes the functionalities and testing methods for proposed voice assistant. The implementation of system takes place in three phases as follows:

5.1Developing the conversational environment

For the development of conversational environment, the first and foremost thing is to make Voice assistant talk, the engine is set to Pyttsx3 which is Python text to speech module and Microsoft Speech API sapi5 uses this for speech synthesis

```
engine = pyttsx3.init('sapi5')
voices = engine.getProperty('voices')
engine.setProperty('voice', voices[0].id)
```

Fig.5.1 Setting voice for assistant

```
def speak(audio):
    engine.say(audio)
    engine.runAndWait()
    print(audio)
```

Fig.5.2 Function to make assistant speak

The next functionality that proposed voice assistants offers, is taking voice command as input with the help of microphone of the system. The function take_command() uses speechRecognition module which supports Google Cloud Speech API for converting speech into text format. A try and except block is also added to the program to handle the errors effectively.

```python
def take_command():
    r = sr.Recognizer()
    with sr.Microphone() as source:
        print("Listening...")
        r.pause_threshold = 1
        audio = r.listen(source)

    try:
        print("Recognizing...")
        query = r.recognize_google(audio, language='en-IN')
        query = query.lower()
        print(f"User said: {query}\n")

    except Exception as e:
        print(e)
        speak("say that again please...")
        return "None"
    return query
```

Fig. 5.3 Function to capture voice command

5.2Developing code for handling commands:

The Python script for handling various user commands is developed. Decision making is required for executing code only if a certain condition is satisfied. Here, in the implemented code condition satisfaction is based on checking if the keyword of the query is present in the code. Execution occurs only when the query is found to be present in the code. Depending upon the functionalities that user want in voice assistant the python script is developed for the same. The functionalities developed for proposed system .

5.3Testing and Debugging

• Functional Testing

Each functionality of the voice assistant is tested with the set of questions to verify that the desired output for each question is obtained.

• Stress Testing

In this testing method the same functionality is tested against all the possible command with minimal keywords that user can pass for getting certain result.

• Validation Testing

The system's ability of choosing most valid or appropriate word is tested against the homophones, the words with same pronunciation but different spelling. This is done to see if system can differentiate between homophones by extracting the context from user's command

CHAPTER 6

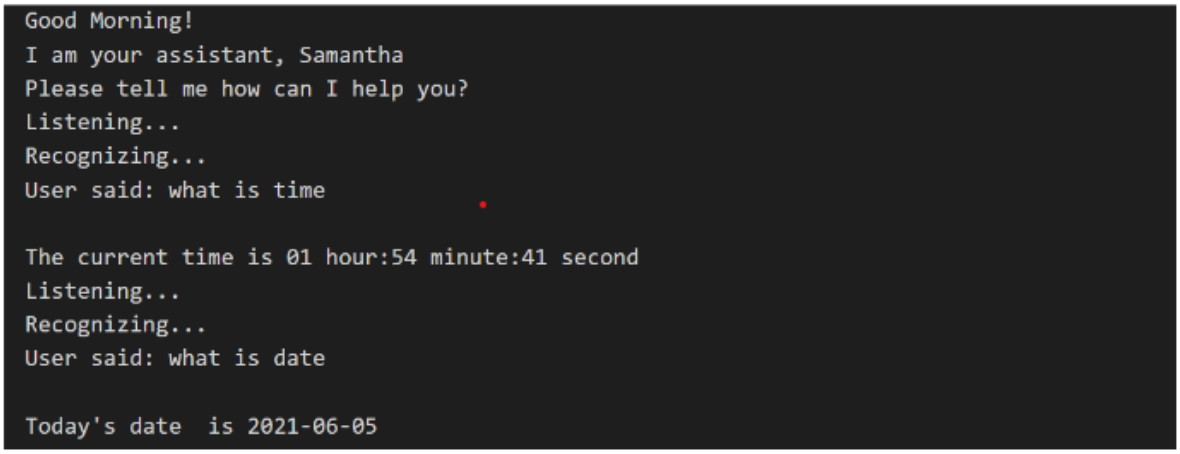
RESULT AND DISCUSSION

In this chapter, the various results of execution are shown. When the systemruns the program, the following results are displayed. All the preparation had been done and the testing was performed according

to the various types of input provided and got outputs. Various inputs lead to the unique outputs.

6.1 Result for Functional Commands:

. Getting Date and Time



In the time functionality current time is displayed and spoken by system in format(HH/MM/SS). Similarly, when date is asked it returned today's date in format (YYYY/MM/DD).

- **Getting Weather Report:**

```
Listening...
Recognizing...
User said: show me weather report

getting the API key
For which city you want me too find the weather?
Listening...
Recognizing...
User said: mumbai

Showing you the weather report
 The temperature in Kelvin is 301.14 The humidity is 78 and The weather description is haze
```

Fig.6.2 Weather functionality

In this functionality, to get the real-time weather update, weather API is used. The API key is used for accessing the weather API account. It again asked user the name of city for which user want to find the weather, after taking it as input the result is both displayed and spoken by voice assistant. The result included temperature, humidity and weather description.

. Getting News Report:

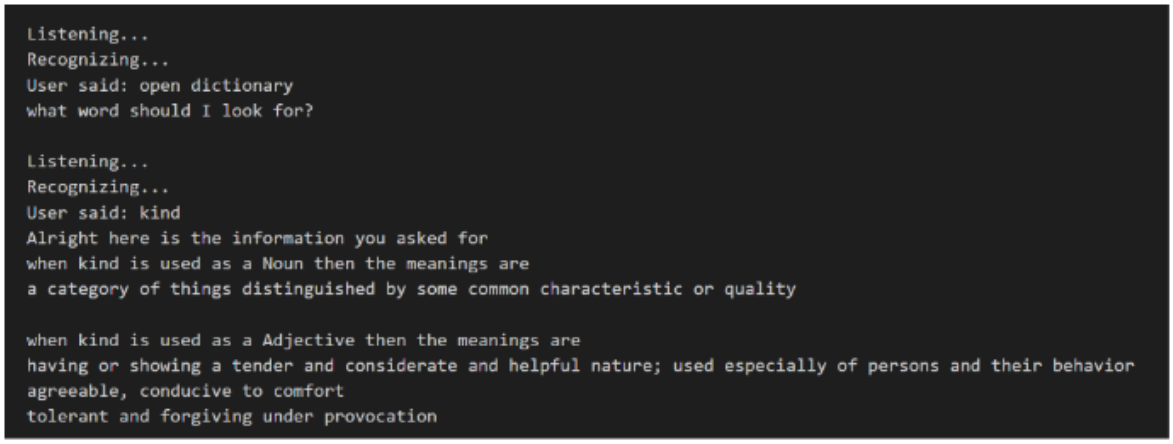
```
Listening...
Recognizing...
User said: show me the news report

taking you to today's news headlines
```

. Getting Current Location:

```
Listening...
Recognizing...
User said: what is my current location
Getting IP address for finding the location
you are currently in Pune city of country India
```

The desktop application Notepad is launched when user commanded. On receiving the user's command to close the launched application, voice assistant closed that.

. **Using Dictionary:**



In the dictionary functionality, the voice assistant responded with the meaning of the required word in its noun form and also in adjective form.

.

• **Solving Calculations:**

```
Listening...
Recognizing...
User said: calculate
what should I calculate

Listening...
Recognizing...
User said: sin x dx
the answer is
integral sin(x) dx = -cos(x) + constant

Listening...
Recognizing...
User said: do calculation
what should I calculate

Listening...
Recognizing...
User said: 897 x 534
the answer is
478998
```

This functionality used the Wolframalpha API for solving the mathematical questions. The API used the key for accessing the account and then returned the result accurately. It solved not only multiplication but also the integral of trigonometric function.

Switching Tabs

```
Listening...
Recognizing...
User said: switch tab
Switching the tab
```

Fig.6.20 Switch tab functionality

The switch functionality switches between the open tabs. On switch command of user, the voice assistant switched from active tab to the next open tab. The user can go back to the previously active tab by switch command again.

• **Shutting Down the System**

```
Listening...
Recognizing...
User said: shutdown the system
```

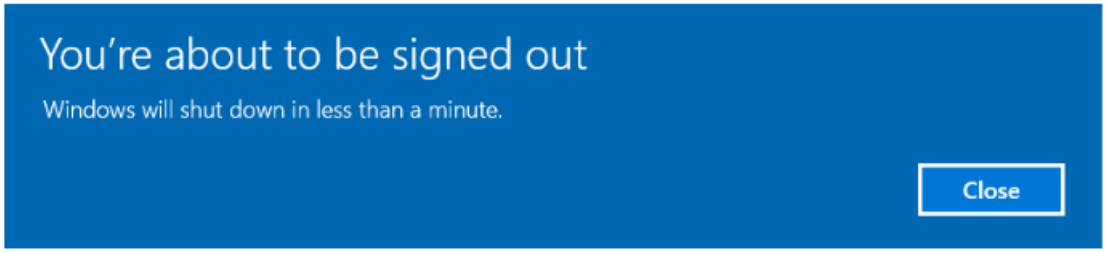


Fig.6.23 Shutdown notification

The assistant first generated a notification before shutting down system. Then within the one-minute system is shut downed.

6.2 Result for Multiple Command Handling

```
Listening...
Recognizing...
User said: my current location
Getting IP address for finding the location
you are currently in Pune city of country India

Listening...
Recognizing...
User said: show my location information
Getting IP address for finding the location
you are currently in Pune city of country India

Listening...
Recognizing...
User said: where i am now
Getting IP address for finding the location
you are currently in Pune city of country India
```

Multiple commands are passed for getting the same accurate result. This is done to ensure that if the system can easily understand what user want to ask even when the user's commands vary for the same function. Here for "my current location", "show my location information" and "where am I now" commands the system accessed the location of user accurately.

6.3 Result for Validation

```
Listening...
Recognizing...
User said: his son studies in university
Depending on context, word 'son' is more valid than 'sun' for this sentence

Listening...
Recognizing...
User said: distance between sun and moon
Depending on context, word 'sun' is more valid than 'son' for this sentence
```

Fig. 6.25 Validation using NLP

Accuracy of speech recognition is main factor in determining the efficiency of voice assistant. It is important to check if system can differentiate between the same soundingwords but having different meanings. So, for ensuring the accurate working it is important to know that if system is really using the NLP to understand the context. The

word "son" and "sun" are homophones, words with same pronunciation and different spelling and meaning. Now when user said "his son studies in university", the system determined the probability of words "son" and "sun", then the word having maximum probability with the given context is selected. The ability to differentiate between the homophones and using most appropriate one based on context, maximizes the accuracy rate of overall system

CHAPTER 7

CONCLUSION

This system is designed in such a method wherein the user accommodates to it effortlessly. The proposed system is implemented using speech recognition and speech synthesis. The proposed intelligent desktop voice assistant throws light on word

recognition problem when homophones are used. It uses Natural Language Processing for recognizing the most appropriate word which user said by understanding the context. This makes system more reliable and easily accessible to user. In beginning,

the current voice assistants and main issues behind it are introduced. After that the related work in field of voice assistant is covered. In the most crucial part, comprehensive amount of study is done about the overall system architecture along with task-oriented and knowledge-oriented approach.

The system also operates in background thus, establishing hands-free interface between user and desktop. Moreover, the system carries out variety of tasks with ease such as telling date and time, playing videos, telling some jokes, getting weather and news report, getting current location, solving calculations, sending e-mails, sending WhatsApp message, finding meaning of any word, answering any user query, switching between the tabs, opening/closing desktop application and even shutting down a user's desktop, just on simple voice command. Thus, it can conclude that the proposed intelligent desktop voice assistant can effectively perform various tasks on voice commands in less time.

7.1 Limitations of the Study

There are three prerequisite that system should meet for smooth work experience.

1. The working microphone is necessary for capturing user's voice command.

2. The stable internet connection is must.

3. The user must have unique API keys for accessing various APIs.

7.2 Future Scope of Work

The voice assistant system in near future can be improved in few ways to make it more usable and accessible.

1. Ensuring Offline Working

At remote places where a reliable internet connection may be not be available all the time, the developed system becomes difficult to be used. To overcome this, the offline speech recognition system can be implemented. This will make system more reliable.

2. Expanding the Scope

The system is currently developed to perform the various tasks depending on user requirement. It can potentially be developed for any sector or business based on their requirements for performing specific tasks more efficiently.

3. Introducing Multiple Language Support

The system is currently usable for the English language. However, this language support can be improved by including the different languages. This will make it possible for user to access the voice assistant in native language giving user more friendly environment..

References

1. Prerna Wadikar, Nidhi Sargar, Rahool Rangnekar, Prof.Pankaj Kunekar (2020) “Home Automation using Voice Commands in the Hindi Language”, International Research Journal of Engineering and Technology (IRJET)

2. Chen-Yen Peng and Rung-Chin Chen (2018) “Voice Recognition by Google Home and Raspberry Pi for Smart Socket Control”, Tenth International Conference on Advanced Computational Intelligence (ICACI)

3. Steve Joseph, Chetan Jha, Dipesh Jain, Saurabh Gavali, Manish Salvi (2020) “Voice based E-Mail for the Blind”, International Research Journal of Engineering and Technology (IRJET)

4. Nishank Tembhurne, Sumedh Vaidya, Afrin Shiekh, Prof. Swapnil Dravyakar (2019) “Voice Assistant for Visually Impaired People”, International Research Journal of Engineering and Technology (IRJET)

5. Shen Hui, Song Qunying and Andreas Nilsson (2012) “Intelligent Voice Assistant”, DiVA Portal

6. Harkishen Singh, Muskan Khedia, Jayashree Panda, Subham Mishra, Ankit Singh (2020) “Jarvis: The Personal Linux Assistant”

7. Rishabh Shah, Siddhant Lahoti, Prof. Lavanya. K (2017) “An Intelligent Chatbot using Natural Language Processing”, International Journal of Engineering Research, Vol. 6, pp. 281-286

8. Win Shih and Erin Rivero (2020) "Virtual Voice Assistants", Library Technology Reports vol. 56, no. 4

9. Khushitha Anand, Prince, Jithin P Sajeevan (2020) “Voice Assistant for Windows”, International Journal for Scientific Research & Development (IJSRD) Vol 8

10. Nikhil Patel, Trupti Landge, Radhika Tiwari, Arjun Verma, Prof. Shabana Pathan (2020) “Desktop Voice Assistant”, International Research Journal of Engineering and Technology (IRJET)

11. George Terzopoulos and Maya Satratzemi (2019) “Voice Assistants and Artificial Intelligence in Education”, The 9th Balkan Conference

12. https://ieeexplore.ieee.org/document/9051160 Accessed on 1st April 2021 at 2:21 p.m.

13. https://towardsdatascience.com/understanding-nlp-how-ai-understands-our- languages-77601002cffc Accessed on 4th April 2021 at 3:42 p.m