

CMR INSTITUTE OF TECHNOLOGY

DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

Title : Virtual Assistant using Python

Presented By :

Keerthi.A(1CR22AI053)

G.Likhitha(1CR22AI041)

Thimmana Gouda Mali Patil(1CR22AI122)

INTRODUCTION

A virtual assistant is a software application that understands natural language voice commands and completes tasks for the user. These tasks can range from setting reminders, sending messages, and answering questions to more complex functions like controlling smart home devices, managing calendars, and providing real-time information on various topics. Virtual assistants use a combination of artificial intelligence, machine learning, and natural language processing to interpret user commands and perform actions.

KEY FEATURES :

1. Voice Recognition:

- Understands and processes spoken commands.

2. Natural Language Processing (NLP):

- Interprets human language to comprehend context and intent.

3. Task Automation:

- Automates routine tasks like setting alarms, sending messages, and managing to-do lists.

4. Information Retrieval:

- Provides quick access to information from online sources.

5. Integration with Services and Devices:

- Controls smart home devices, integrates with calendars, and uses third-party services.

6. Learning and Personalization:

- Learns user preferences to offer personalized responses and recommendations.

Examples of Popular Virtual Assistants

- **Amazon Alexa:** Integrated with Echo devices, Alexa can control smart home devices, play music, answer questions, and more.
- **Google Assistant:** Available on Android devices and Google Home, it offers comprehensive information retrieval, task automation, and smart home control.
- **Apple Siri:** Integrated into iOS devices, Siri can send messages, set reminders, provide directions, and interact with other Apple services.
- **Microsoft Cortana:** Available on Windows devices, Cortana helps with task management, reminders, and information retrieval.

This code creates a voice-activated assistant using Python, utilizing several libraries to handle speech recognition, text-to-speech, and interaction with online services. Here's an explanation of how the code works:

Import Libraries:

- `speech_recognition` (alias `sr`) for recognizing speech.
- `pyttsx3` for text-to-speech conversion.
- `pywhatkit` for playing YouTube videos.
- `datetime` for getting the current time and day.
- `time` for delaying operations.
- `wikipedia` for fetching summaries from Wikipedia.

Initialize Recognizer and Text-to-Speech Engine:

- `r = sr.Recognizer()` initializes the recognizer.
- `machine = pyttsx3.init()` initializes the text-to-speech engine.

talk Function:

This function uses the text-to-speech engine to speak any given text.

get_instruction Function:

This function listens for a voice command, processes it to remove the keyword "john", and returns the cleaned command. It handles exceptions for unrecognized speech and request errors.

play_instruction Function:

This function processes the instruction and takes appropriate action:

- Play Command: Plays a song on YouTube.
- Time Command: Announces the current time.
- Day Command: Announces the current day.
- Stop Command: Stops the program loop.
- Information Command: Retrieves and announces a Wikipedia summary.
- Handles unknown or invalid instructions.

Main Loop:

The main loop keeps the assistant running, continuously listening for and processing commands until the "stop" command is received.

CODE :

```
terminal Help C:\> Creating virtual assistant > john.py > ...
john.py X
1 import speech_recognition as sr
2 import pyttsx3
3 import pywhatkit
4 import datetime
5 import time
6 import wikipedia
7
8 # Initialize the recognizer and the text-to-speech engine
9 r = sr.Recognizer()
10 machine = pyttsx3.init()
11
12 def talk(text):
13     """Speak the given text."""
14     machine.say(text)
15     machine.runAndWait()
16
17 def get_instruction():
18     """Listen for a voice command and return it."""
19     try:
20         with sr.Microphone(device_index=0) as source:
21             print("Listening...")
22             r.adjust_for_ambient_noise(source)
23             speech = r.listen(source)
24             instruction = r.recognize_google(speech)
25             instruction = instruction.lower()
26             print("Instruction heard: ", instruction)
27             if "john" in instruction:
28                 instruction = instruction.replace('john', '').strip()
29                 return instruction
30     except sr.UnknownValueError:
31         print("Sorry, I did not understand that.")
32     except sr.RequestError as e:
33         print("Could not request results from Google Speech Recognition service; {}".format(e))
34     return None
35
```

```
john.py X
C:\> Creating virtual assistant > john.py > ...
36 def play_instruction():
37     """Process the instruction and take action based on it."""
38     instruction = get_instruction()
39     if instruction:
40         print("Processed instruction: ", instruction)
41         if "play" in instruction:
42             song = instruction.replace('play', '').strip()
43             talk("Playing " + song)
44             pywhatkit.playonyt(song)
45             # Wait for some time to ensure the command is executed
46             time.sleep(10)
47         elif 'time' in instruction:
48             time_now = datetime.datetime.now().strftime('%I:%M %p')
49             talk('The current time is ' + time_now)
50             print('The current time is ' + time_now)
51         elif 'day' in instruction:
52             day_now = datetime.datetime.now().strftime('%A')
53             talk('Today is ' + day_now)
54             print('Today is ' + day_now)
55         elif 'stop' in instruction:
56             talk('Stopping the program')
57             return False # Signal to stop the loop
58         elif 'information' in instruction:
59             query = instruction.replace('information', '').strip()
60             talk('Searching for information about ' + query)
61             try:
62                 summary = wikipedia.summary(query, sentences=2)
63                 talk(summary)
64                 print("Information about " + query + ": " + summary)
65             except wikipedia.exceptions.DisambiguationError as e:
66                 talk('There are multiple results for this query. Please be more specific.')
67                 print('There are multiple results for this query. Please be more specific.')
68             except wikipedia.exceptions.PageError:
69                 talk('No information found for ' + query)
```

```

70         print('No information found for ' + query)
71     else:
72         talk("I didn't understand that command.")
73         print("I didn't understand that command.")
74     else:
75         talk("No valid instruction received.")
76         print("No valid instruction received.")
77     return True # Continue the loop
78
79 # Main loop to keep listening for commands
80 if __name__ == "__main__":
81     listening = True
82     while listening:
83         listening = play_instruction()
84

```

OUTPUT :

```

OUTPUT    DEBUG CONSOLE    TERMINAL    PROBLEMS    PORTS

Active code page: 65001

C:\Users\keert\mern6>python -u "c:\Creating virtual assistant\john.py"
Listening...
Instruction heard:  john what time is it
Processed instruction:  what time is it
Listening...
Instruction heard:  on what day is it
Listening...
Instruction heard:  on what day is it
Listening...
Instruction heard:  what day is it
Listening...
Instruction heard:  on what day is it
Listening...
Instruction heard:  john what day is it
Processed instruction:  what day is it
Listening...
Instruction heard:  john stop the program
Processed instruction:  stop the program

C:\Users\keert\mern6>python -u "c:\Creating virtual assistant\john.py"
Listening...
Instruction heard:  john give me information about ai
Processed instruction:  give me information about ai
Listening...
Instruction heard:  stop the program
Listening...
Instruction heard:  john stop the program
Processed instruction:  stop the program

C:\Users\keert\mern6>python -u "c:\Creating virtual assistant\john.py"
Listening...
Instruction heard:  john play vande mataram
Processed instruction:  play vande mataram
Listening...
Sorry, I did not understand that.
No valid instruction received.
Listening...
Sorry, I did not understand that.
No valid instruction received.
Listening...
Sorry, I did not understand that.

Ln 84, Col 1  Spaces

```

```
No valid instruction received.
Listening...
Sorry, I did not understand that.
No valid instruction received.
Listening...
Sorry, I did not understand that.
No valid instruction received.
Listening...
Instruction heard: john give me information about a
Processed instruction: give me information about a
Information about give me about a: "Give In to Me" is a song by American singer-songwriter Michael Jackson, released as the seventh single from his eighth studio album, Dangerous (1991). Released in February 1993 by Epic Records, the song peaked at number one in New Zealand for four consecutive weeks and at number two on the UK Singles Chart.
Listening...
Sorry, I did not understand that.
No valid instruction received.
Listening...
Instruction heard: don't give me information about ai
No valid instruction received.
Listening...
Instruction heard: john give me information about
Processed instruction: give me information about
Information about give me about: "Give In to Me" is a song by American singer-songwriter Michael Jackson, released as the seventh single from his eighth studio album, Dangerous (1991). Released in February 1993 by Epic Records, the song peaked at number one in New Zealand for four consecutive weeks and at number two on the UK Singles Chart.
Listening...
Instruction heard: john give me information about ai
Processed instruction: give me information about ai
Information about give me about ai: Character.ai (also known as c.ai or Character AI) is a neural language model chatbot service that can generate human-like text responses and participate in contextual conversation. Constructed by previous developers of Google's LaMDA, Noam Shazeer and Daniel De Freitas, the beta model was made available to use by the public in September 2022.
Listening...
Instruction heard: john information on artificial intelligence
Processed instruction: information on artificial intelligence
Information about on artificial intelligence: Artificial intelligence (AI), in its broadest sense, is intelligence exhibited by machines, particularly computer systems. It is a field of research in computer science that develops and studies methods and software that enable machines to perceive their environment and use learning and intelligence to take actions that maximize their chances of achieving defined goals.
Listening...
Instruction heard: john stop the program
Processed instruction: stop the program
```

CONCLUSION

This virtual assistant script uses Python to enable voice command recognition, task execution (such as playing music and fetching information), and provides feedback to the user. It's a basic implementation that could be expanded to support more commands and integrate with additional services for enhanced functionality and user interaction.