

PROJECT REPORT ON Self Learning Bot REVIEW - II

Student Details

| <u>Roll No</u> | <u>Name</u> |
|----------------|-------------------|
| 20201CAI0104 | Maheswar Reddy |
| 20201CAI0131 | Karthik Sharma |
| 20201CAI0152 | Arun Teja |
| 20201CAI0096 | Salapakshi Ganesh |

Under the Supervision of :

Prof . Dr.Asif Mohammed H.B

December , 2023

**DEPARTMENT OF COMPUTER SCIENCE AND
ENGINEERING
PRESIDENCY UNIVERSITY
BENGALURU**

Abstract:

This report outlines the current voice-activated AI assistant program. It suggests an evolution towards a more comprehensive training model that integrates technical and non-technical modules for a more well-rounded training experience.

Introduction:

The provided script currently implements a voice-controlled assistant using Python with various libraries such as `pyttsx3`, `speech recognition`, `Threading`, `google-trans`, `Selenium`, and custom modules like `toKnow`, `Review`, and `trans`, `google`. In the future, the script aims to enhance its capabilities through the integration of Natural Language Processing (NLP) for semantic analysis using libraries like SpaCy. The envisioned improvement involves training the system to provide accurate answers to both technical and non-technical questions.

Pseudo Code:

Main.py:

- Initialize text-to-speech engine
- Initialize speech recognition engine

Greet user

- Listen for the initial response
- Recognize and print the response

If the response contains "good," "you," or "fine":

Respond positively and ask for the user's command

Infinite loop:

- Listen for the user's command
- Recognize and print the command

If a command contains "data," "something," or "about this":

- Ask for specific information
- Listen for the information
- Recognize and print the information
- Search for information in Wiki using the toKnow module
- Ask if the user needs something else

Else, if the command contains "product," "review," or "video":

- Ask for the product name
- Listen for the product name
- Recognize and print the product name
- Play the product review on YouTube using the Review module
- Ask if the user needs something else

Else, if the command contains "translate" or "language":

Invoke translation function using trans()

Else, if the command contains "Bye" or "See you later":

Bid farewell and exit the loop

Exit the program

review.py:

Class: Review

Method: `_init_`

Initialize the Review class

Create a new instance of the Chrome WebDriver

Method: `get_review(query)`

- Set the query attribute to the provided query
- Navigate to the YouTube search results page for the given query
- Use `WebDriverWait` to wait for the first video thumbnail to be clickable
- Click on the clickable thumbnail
- Handle any exceptions that may occur during the process

Example Usage:

Create an instance of the Review class

```
ob = Review()
```

Call the `get_review` method with a specific query (e.g., "product review")

```
ob.get_review("product review")
```

todoautomation.py:

Class: toKnow

Method: `getinfo`

Initialize the toKnow class

Create a new instance of the Chrome WebDriver

Method: get_info(query)

- Set the query attribute to the provided query
- Navigate to the Wikipedia homepage
- Find the search input element on the page
- Click on the search input element
- Enter the provided query into the search input
- Find the enter/search button on the page
- Click the enter/search button
- Pause script execution for 60 seconds (you may want to replace this with a more dynamic wait)

Example Usage:

Create an instance of the toKnow class

```
ob = toKnow()
```

Call the get_info method with a specific query (e.g., 'Emma Watson')

```
ob.get_info('Thomas Alva Edison')
```

translate.py:

Import required libraries:

- google_trans
- Translator from google_trans
- sr from speech_recognition
- Translator from googletrans
- gTTS from gtts
- playsound from playsound
- os

Print supported languages using google_trans.LANGUAGES

Define a translation function, trans():

- Create a Microphone instance mic
- Create a Recognizer instance r

Inside a context manager with the mic as a source:

- Create a Translator instance translator
- Set input_language to from language
- Set output_language to language which you wanna to translate
- Print a message asking the user to speak the text for translation
- Adjust ambient noise for 0.2 seconds
- Listen for audio using r.listen(source)
- Recognize the audio to obtain text3 using r.recognize_google(audio3)
- Print the recognized text
- Translate the text3 from input language to output language using translator.translate()
- Get the translated text (transed_text) from the translation result
- Print the translated meaning
- Create a gTTS instance named speak with the translated text, output language, and slow set to False
- Save the gTTS instance as "captured_voice.mp3"
- Play the saved audio using playsound('captured_voice.mp3')
- Remove the temporary audio file 'captured_voice.mp3'
- Print the translated text

Call the trans() function

Current Features:

1. Voice Interaction:

- Utilizes the `speech recognition` library to recognize and interpret voice commands.
- Employs the `pyttsx3` library for text-to-speech functionality, enabling audible responses.

2. Functionality:

- Interacts with users through voice, responding to greetings and inquiries.
- Executes actions based on recognized commands such as opening wiki for info you need, searching google to search what you want, translation helps the assistant to assist you in your own Native Language, and finding and playing the reviews for your favourite product in YouTube.

3. Code Structure:

- Organized into functions and classes, promoting modularity and reusability.
- Each major functionality is encapsulated in separate modules or methods within the main script.

4. User Interaction:

- Engages users conversationally, asking for input and responding appropriately.

- Implements a loop for continuous listening and action execution.

5. Selenium Integration:

- Utilizes the Selenium library for web automation in class like toKnow() and Reviews().

6. Translation:

- Incorporates translation functionality using Google Translate and recognizing the audio and converting to text and then speaking it you as you Native speaker .

Future Enhancements:

1. NLP Integration:

- Incorporation of NLP libraries like SpaCy for semantic analysis.
- Training the system to understand and respond accurately to both technical and non-technical questions.

2. Improved Understanding:

- Enhanced understanding of user queries through semantic analysis.
- Better differentiation between technical and non-technical terms for more precise responses.

3. Training Data:

- Gathering and curating training data comprising both technical and non-technical questions.
- Training the NLP model to improve accuracy in recognizing and responding to various query types.

4. User Feedback Loop:

- Implement a feedback loop for users to provide feedback on the accuracy of responses.

References:

- <https://www.tutorialspoint.com/selenium/index.htm>
- <https://py-googletrans.readthedocs.io/en/latest/>
- https://cloud.google.com/translate/docs/basic/translate-text-basic#translate_text-python

Conclusion:

The planned integration of NLP with libraries like SpaCy represents a significant advancement for the voice-controlled assistant script. With improved semantic analysis and training on a diverse dataset, the assistant aims to provide more accurate and context-aware responses to both technical and non-technical queries, enhancing the overall user experience.

THANK YOU