

TypeBot: A text based Chatbot

Problems Faced:

Generally web applications have certain helpline numbers or chats that a certain user can use to communicate the issues they might be facing. In such a case, if a user wants some help at a time which is beyond the common office hours, he might not get an immediate solution to his problem. Either he is asked to call the next day or just put a message on the chat and expect a reply only the next day. While the user is considering something to be bought but is looking for its features, he/she might not find personal and manual assistance.

Project Idea: TypeBot

TypeBot is a text-based chatbot. It is a python based project which allows the user to talk to the system. I had initially implemented a voice-enabled chatbot. As the linux OS on my system couldn't access the microphones, I implemented a text-based chatbot. TypeBot can search the web whatever you want to ask (using wikipedia). It can go. Or else it could just talk to you in normal ways. It can tell you the current date and time.

Tech Stack

Implementation of the Voice-enabled Chatbot

The voice enabled chatbot is written in Python. It uses Python's text to speech library Pyttsx3. It uses the Speech Recognition package of Python which contains the Google's Web Speech API to accept the user's voice input. This project is hosted by Python's Flask server.

Implementation Of the Text-Enabled Chatbot

This uses the webbrowser to access the Internet and wikipedia to search information asked by the user. I used the PyOWM API to predict the weather. The user's input is taken using the HTML form and parsed to the flask server using POST method request. This parsed string is now passed to the run() method of the Python Backend file. The Backend processes the data and returns a response to the Flask server which in turn parses it to the Frontend HTML using Jinja2.

Progress As Of Now

I had implemented the voice-enabled chatbot earlier. I have uploaded the code on GitHub too. The Linux OS on my system could not access microphones and hence, I wasn't able to run it on my system. Hence, I then implemented a text-based chatbot which takes user messages and processes them to give a reply. It can open websites using the webbrowser. It can search for something which the user enters using wikipedia. It can tell the user the current date and time. It can tell you about today's weather.

Future Ideas

- I think of implementing a task manager with the Flask app. It will record the user's To-Do list.
- I also want to get the voice-enabled chatbot working with the same features as the text based chatbot.
- I also want to implement a Birthday wishing part where wishing a person for his/her birthday is automated.
- I also want to find a way to save and record the user's searches so that his search

next time would be made easier.

References

- Flask Tutorials
- CodeMentor
- Stackoverflow
- askUbuntu (for debugging system issues such as the ALSA script configuration for voice-enabled chatbot)