FAQ Bot Documentation

Project files are divided into text files and Python files.

Text Files

There are 4 text files.

questions.txt

It contains the questions that the bot can answer. They are separated by lines and the last character is "?".

required.txt

It contains the words necessary for the bot to answer the correct question. They must be the most different from others and essential for the question to be understood.

answers.txt

It contains the answers that the bot will respond exactly as written. They are separated by lines and correspond to the respective line of the questions.txt file.

bot_token.txt

It contains the bot token to be able to get the connection with discord.

Python Files

It contains 3 supporting and 2 main files (test and with discord).

greetings.py

Greetings, goodbyes, thanks and compliments are defined as how the bot can answer. They are defined as string and then converted to JSON.

functions.py

It contains 5 functions. The first 3 functions are to load the data from the answers.txt, questions.txt and required.txt files. Answers return as a list of phrases, questions as lists of words, and those required as lists of words as well.

The *phase3* function obtains user input and in the event of an incompatibility with the registered questions, the function will seek a more user-friendly answer. Implying

that you understand that you are talking about an organization, location or a person. In case of none, you will answer not knowing about it with the *random_string* function that complements *phase3*.

load_data.py

Imports the *greetings.py* and *functions.py* files to make use of the data and functions that are defined there.

The first part (*load_json*), defines the function that will join and convert the questions, answers and required to convert them into a string and in turn, join it with the greetings defined above and in that format.

The next part, gets the questions, answers and required to join them in with the function *load_json* which a string returns and then use the load method of the json library (*json.load()*) to convert the resulting string into JSON.

Finally, the function that will receive the user's input is defined and that will choose the response that will be returned by the bot (*get_response*). This function separates the input it receives from the user in a list of words giving importance to the question character "?". This list is compared with the list of required words and the one with the highest number of matches will be used as the answer. If you don't have any required words, you'll compare with the whole questions and if you can't find it, you'll give an answer using the *phase3* function.

faq_bot_.py

Imports discord libraries, sys to finish program and *load_data.py* functions.

First function is to obtain the response and send the message to Discord.

Second function is to talk with the bot. Print the initial message in the terminal and do the connection with Discord reading the token. 2 events are created, one to announce that the bot is connected with discord (you can also check if the bot is online on the channel itself).

The second event is to create the interaction between the user and the bot in the discord channel. You can also interact with it by direct message. The end of the conversation occurs when the user says goodbye to the bot.

test.py

This file imports the functions of load_data.py and executes a test function to interact with the bot only from the program terminal. This program ends up running when the user says goodbye to the bot as well.