Chatbot Using Python

Chatbot.py

```
#Python Chatbot by Junaid Khateeb

import re

import long_responses as long

#long_responses is another python file created by us,it is not a default one. YOu will need to have that too in the same location as the main program.


def message_probability(user_message, recognised_words, single_response=False, required_words=[]):

    message_certainty = 0

    has_required_words = True


    # Counts how many words are present in each predefined message

    for word in user_message:

        if word in recognised_words:

            message_certainty += 1


    # Calculates the percent of recognised words in a user message

    percentage = float(message_certainty) / float(len(recognised_words))


    # Checks that the required words are in the string

    for word in required_words:

        if word not in user_message:

            has_required_words = False

            break
```

```python
    # Must either have the required words, or be a single response

    if has_required_words or single_response:

        return int(percentage * 100)

    else:

        return 0




def check_all_messages(message):

    highest_prob_list = {}



    # Simplifies response creation / adds it to the dict

    def response(bot_response, list_of_words, single_response=False, required_words=[]):

        nonlocal highest_prob_list

        highest_prob_list[bot_response] = message_probability(message, list_of_words, single_response, required_words)



    # Responses -----------------------------------------------------------------------------------

    response('Hello and Namaskar', ['hello', 'hi', 'hey', 'sup', 'heyo','hola','wassup','hii'], single_response=True)

    response('See you!', ['bye', 'goodbye'], single_response=True)

    response('I\'m doing fine, and you?', ['how', 'are', 'you', 'doing'], required_words=['how'])

    response('You\'re welcome!', ['thank', 'thanks'], single_response=True)

    response('Thank you!', ['i', 'love', 'python', 'coding'], required_words=['python'])
```

```python
    # Longer responses

    response(long.R_ADVICE, ['give', 'advice'], required_words=['advice'])

    response(long.R_EATING, ['what', 'you', 'eat'], required_words=['you', 'eat'])

    response(long.R_TRAINING, ['you', 'like', 'training', 'course'], required_words=['like'])

    response(long.R_TRAINER, ['who', 'trainer', 'teacher', 'coach'], required_words=['who'])


    best_match = max(highest_prob_list, key=highest_prob_list.get)

    # print(highest_prob_list)

    # print(f'Best match = {best_match} | Score: {highest_prob_list[best_match]}')


    return long.unknown() if highest_prob_list[best_match] < 1 else best_match



# Used to get the response

def get_response(user_input):

    split_message = re.split(r'\s+|[,;?!.-]\s*', user_input.lower())

    response = check_all_messages(split_message)

    return response


#codes by Junaid Khateeb

# Testing the response system

while True:

    print('Bot: ' + get_response(input('You: ')))



long_responses.py

import random
```

R_EATING = "I don't like eating anything because I'm a bot obviously!"

R_ADVICE = "If I were you, I would go to the internet and type exactly what you wrote there!"

R_TRAINING = "Oh, I am loving the training"

R_TRAINER = "Junaid Khateeb from Mumbai is the trainer on this course, do check www.khateebstudyabroad.com"

```python
def unknown():
    response = ["Could you please re-phrase that? ",

        "...",

        "Sounds about right.",

        "What does that mean?"

        "Sorry, I am unable to understand that,I am still a young bot"][
    random.randrange(4)] #randrange cvalue has to be equal to the number of responses that we have used.

    return response
```

#codes by Junaid Khateeb