COSC 101 Homework 7: Spring 2025

Introduction

ELIZA was an early chatbot that used pattern matching and string substitution to mimic a therapist. In this homework you will implement a version of ELIZA. To do this, you will use the following string and list concepts:

- Indexing
- Slicing
- Pattern matching

You will also continue to practice with - developing modular application - using `doctests` for test-driven development and to cover many edge cases and - leveraging from bottom-up implementation and top-down design.

Your assignment

Your task is to complete the following steps:

1. Download the `hw6.zip` file and open it. You will see one python file: `hw7_eliza.py` in the unzipped folder.
2. Write and test 12 helper functions described below first and then write the `main` function to run our Eliza's chat-bot.
3. Interact with the chatbot at least twice. Reflect on the cases in which the chatbot's responses are convincing and cases where its not.
4. Submit your completed program and text files with your interactions.

Required functionality

Your chatbot should have the following functionality.

1. Prints "`ELIZA: Hello, I am Eliza. How are you today?`"
2. Prompts the user for a string with the message "`YOU:`"
3. Prints a response to the user input
4. Repeats steps 2 and 3, until the user types in "`bye`" or "`Bye`".
5. Prints `"Bye! Thanks for chatting!"`

Example output

```
ELIZA: Hello, I am Eliza. How are you today?
YOU: I am sad
ELIZA: I am sorry to hear that. Why do you think you are sad?
YOU: I am always so disappointed in myself
ELIZA: Can you think of a specific example?
YOU: Like I was disappointed because I did not get perfect scores
ELIZA: Can you reflect on why you were disappointed because you did not get perfect scores?
YOU: I want to be perfect all the time
ELIZA: In what way?
YOU: Well I do not want to make any mistakes..
ELIZA: Is it important to you that you do not want to make any mistakes?
YOU: Yes!
ELIZA: Can you expand on that?
YOU: I hate the idea of showing my weak side.
ELIZA: Why do you hate the idea of showing your weak side?
YOU: Bye
ELIZA: Bye! Thanks for chatting!
```

Your tasks

Task A: Implement the helper functions

The starter file comes with `doctests` for each function that explicitly specify input-output mappings for the function. In order to make it easier to implement the functions one at a time, all of the function definitions except the first function are commented out. Implement the first function, ensure that it passes all the doctests, before commenting out the next function.

Each helper function should be very short. Feel free to add your own fruitful functions, each of which abstracting a process that you find repeated across the helper function you need to implement.

Note: In writing some of the helper functions, you might have to compose previously written (and tested!) functions

Implement the helper functions in the following order.

- `uncapitalize(s)`: Turns the first letter of the string sentence into lower case.
- `replace_pronoun(s)`: Changes pronouns in the string sentence from first person to second person: “`I`” becomes “`you`”, “`my`” becomes “`your`”, and “`me`” becomes “`you`”.
- `remove_fillers(s)`: Removes filler words like “`well`”, “`because`” and “`like`” from the beginning of the sentence.
- `remove_punctuation(s)`: Removes punctuation (“`.`”, "`,`", “`!`”, “`?`”) from the end of sentence.
- `one_word(s)`: If the sentence has only one word return “`Can you expand on that?`”
- `sad(s)`: If the sentence `s` is “`I am sad`” or “`I am depressed`”, return respectively the string “`I am sorry to hear that. Why do you think you are sad?`” or “`I am sorry to hear that. Why do you think you are depressed?`”
- `all(s)`: If a sentence has the word “`all`” in it, return “`In what way?`”
- `always(s)`: If a sentence has the word “`always`” in it, return “`Can you think of a specific example?`”
- `hate(s)`: If the pattern “`hate X`” or “`dislike X`” occurs in `s`, return respectively “`Why do you hate X?`” or “`Why do you dislike X?`” (where X can be replaced by any string)
- `am(s)`: If the pattern “`am X`” occurs in the sentence, return “`Can you reflect on why you are X`”. If the pattern “`was X`” occurs instead, return “`Can you reflect on why you were X`”.
- `generic(s)`: Assuming none of the other patterns match, this function is used to return the generic responce “`Is it important to you that X`” where X is `s`, the input sentence.
- `generate_response(s)`: This function combines all other functions and returns a final response.

Task B: Compose the helper functions

Once you have implemented, tested and debugged each these 12 functions, write the `main` function that implements the required functionality.

Task C: Interact with the chatbot

Interact with the chatbot at least twice. Each interaction should have at least 20 inputs from you (i.e., YOU should occur at least 20 times before “bye”). Based on your interactions, reflect on the following question:

1. If you did not know how the chatbot worked, would you be convinced that you were speaking to a real human? Why or why not? Point to concrete examples in the chat!
2. What could you do to make the chatbot more convincing?

Submission

You must submit

- `hw7_eliza.py` with the header completed, including the reflection
- Completed `reflection.pdf` in which you copy-paste your two interactions with the bot and answer the reflection question.