Mr. Hungry Chatbot

Advanced Programming

C-CS219

Student Name	Student ID	Group
Ahmed Yasser Hassanein	21-101006	1
Abdulrahman Khaled Salah	21-101079	1
Omar Mohamed El Abasery	21-101173	1

Contents

Introduction

What is a Chatbot?

A chatbot is a computer program that simulates and processes human conversation (either written or spoken), allowing humans to interact with digital devices as if they were communicating with a real person.

Historic Outline

Chatbots are not as old as they seem to be; the first semi-chatbot, The Turing Test, was developed in 1950. The Turing test was developed by Alan Turing. It was a test of a machine's ability to exhibit intelligent behaviour equivalent to, or indistinguishable from, that of a human. Eliza, the first chatbot, was created by Joseph Weizenbaum, designed to be a therapist. It used to simulate a conversation by using a "pattern matching" and substitution methodology that gave users an illusion of understanding on the part of the bot. Pattern Matching? Phenomenal! Other chatbots were developed along the way; check the "References" section for more information.

Importance of Chatbots

Chatbots boost operational efficiency and bring cost savings to businesses while offering convenience and added services to internal employees and external customers. They allow companies to easily resolve many types of customer queries and issues while reducing the need for human interaction.

How do Chatbots Work?

A chatbot is an automated conversational AI that pretends to be human and carries out programmed tasks based on specific triggers, responding through a web or mobile app. These triggers can be defined as "Pattern Matching" in Functional Programming terms.

Types of Chatbots

Task-oriented (declarative) chatbots are single-purpose programs that focus on performing one function. Using rules, NLP, and very little ML, they generate automated but conversational responses to user inquiries. Interactions with these chatbots are highly specific and structured and are most applicable to support and service functions—think robust, interactive FAQs.

Data-driven and predictive (conversational) chatbots are often referred to as virtual assistants or digital assistants, and they are much more sophisticated, interactive, and personalized than task-oriented chatbots. These chatbots are contextually aware and leverage natural-language understanding (NLU), NLP, and ML to learn as they go. They apply predictive intelligence and analytics to enable personalization based on user profiles and past user behaviour.

Motivation

The food industry/domain was our choice for this project because we saw many individuals, including us, struggling in their choice of "What will I eat on my free day?" We found that it is a tough choice, and we had to find a solution that may be deployed for a price in the future.

Analysis and Design

Modules

All of the following are ".ml" modules that are the building blocks for the main.ml

Rules: A module that contains all the rules that the chatbot uses to validate input and generate a personalized response.

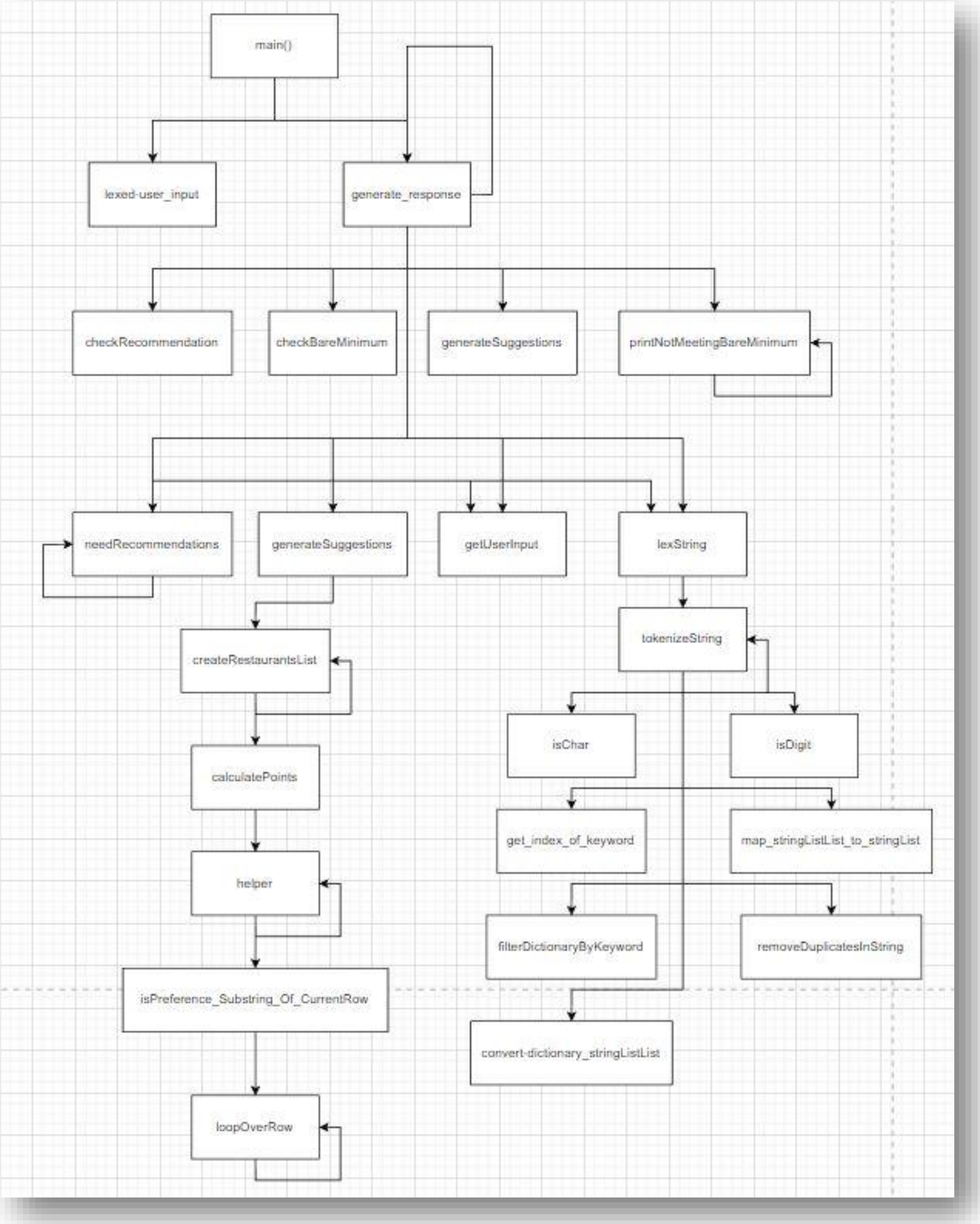
Dictionary: It contains all functions that read from the "Dictionary.csv" file and convert what it's read to suitable data structures that will aid in the Lexing process.

Lex: It contains all functions related to lexing and tokenizing the user input.

Validation: A module that contains all functions related to validating the input and displaying an output of whether the chatbot has understood the input or not.

Restaurant: It contains all functions that read from the "Resaturant.csv" file and convert what it's read to suitable data structures that will aid in giving a personalized recommendation to the user.

Hierarchy Chart



Description

Theory

The method we chose to build upon our project was to choose a set of rules that the chatbot had to validate, chat with the user about, and recommend a restaurant based upon it. Pattern Matching and "while" loops, to an extent, were our dearest friends in building the basis for this code because they helped in the lexing, input validation, and mapping functions.

Explanation of Rules

The user goes into multiple paths depending on his/her input. If the user inputs a "recommendation" keyword that suggests he/she want a recommendation right away, then the input must meet a bare minimum which is the "Cuisine", "Restaurant Location", and "Price Range." When the bare minimum is met, then the user can receive the recommendation. If the user does not require a recommendation, then he/she must answer more questions to personalize the possible recommendation. There are limited questions because the scope of the domain is infinite, and we made it easy to add more questions and rules for a more personalized recommendation.

Workload Distribution

Ahmed Yasser:

- Updated the “Dictionary.csv” file.
- Modularized the main source file to separate “.ml” files.
- Updated the rules to be of type options.
- Created the “resultArray” to help store all user preferences to be able to generate accurate responses.
- Created the “generate_response” function that will ask the user specific questions, and then, decide on which path he/she will go through.
- Created the helper functions in the “generate_response” function.

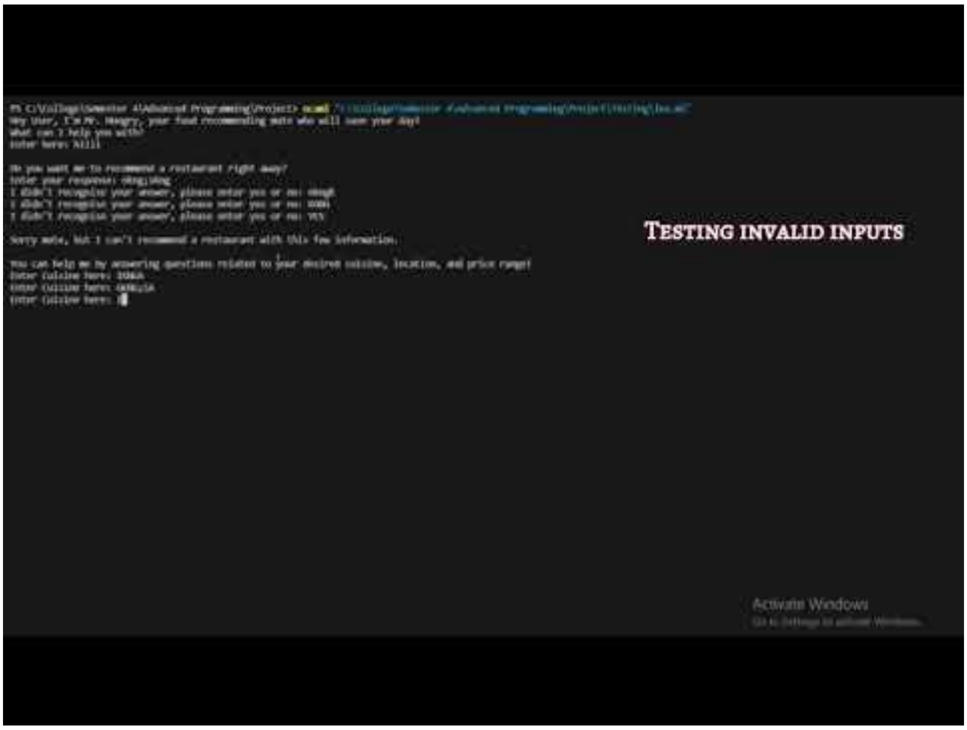
Abdulrahman Khaled:

- Created the “Dictionary.csv” file.
- Created the functions that map the “Dictionary.csv” file data to suitable data structures that will generate personalized responses.
- Set the basic rules of the chatbot and created the rules variant.
- Created the “lexString” function with all of its helper functions.
- Created the “printResponse” function.
- Created the building blocks of the chatbot including its rules, path, greetings, etc.

Omar Elabasery:

- Created the “Restaurant.csv” file.
- Created the functions that map the “Restaurant.csv” file data to suitable data structures to lex the input.
- After mapping the data from the “Restaurant.csv” to suitable data structures to lex the input, he mapped the lexed input to match the “Restaurant.csv” data and generate a personalized response. (found in the “validation.ml” and “restaurant.ml” files)
- Created the “generateSuggestions” function with all of its helper functions.

Demo Video



```
PS C:\College\Semester 4\Advanced Programming\Project> ocaml "c:\College\Semester 4\Advanced Programming\Project\testing\[illegible].ml"
Hey User, I'm Mr. Hungry, your food recommending mate who will save your day!
What can I help you with?
Enter here: hiiii

Do you want me to recommend a restaurant right away?
Enter your response: [illegible]
I didn't recognise your answer, please enter yes or no: [illegible]
I didn't recognise your answer, please enter yes or no: NOOO
I didn't recognise your answer, please enter yes or no: YES

Sorry mate, but I can't recommend a restaurant with this few information.

You can help me by answering questions related to your desired cuisine, location, and price range!
Enter Cuisine here: [illegible]
Enter Cuisine here: [illegible]
Enter Cuisine here: 
```

TESTING INVALID INPUTS

Video Link: https://youtu.be/2TMz09A-Zv0

Challenges & Problems

The challenges we faced were that we researched for days how to implement a Graphic User Interface using OCaml, and we found out that the libraries that support the GUI required either MacOS or Linux operating systems. Another problem was that we did not know when was the chatbot "enough" because we were on a continuous adventure of expanding the domain, meeting many requirements, handling various inputs, and adding several rules. The final challenge we faced was that we did not carefully outline our rules, and we had to continuously change the structure of our code to meet our high expectations that were mentioned in the previous sentence.

Conclusion

We are proud of our work because we did so much in no time, and we were upset that we did not have the privilege to showcase our chatbot in a GUI. Also, we are proud of improving our debugging skills because dealing with OCaml's is another level of nonsense. This project gave light to us on how a chatbot may work, insight on its types, and how we can make a better implementation in the near future.

References

[1] Oracle, “What is a Chatbot?,” *www.oracle.com*, 2022. https://www.oracle.com/chatbots/what-is-a-chatbot/

[2] vivek Patil, “Timeline of Chatbots,” *Medium*, Feb. 23, 2020. https://medium.com/@vivekpatil647/timeline-of-chatbots-f3baf14c05e6

[3] Anil Madhavapeddy and Y. Minsky, *Real World OCaml: Functional Programming for the Masses*. Cambridge University Press, 2022.

[4] S. Pérez-Soler, J. De Lara, U. Autónoma De Madrid, S. Perezs, and E. Guerra, “Model-driven chatbot development.” Accessed: May 22, 2023. [Online]. Available: https://miso.es/pubs/ER20.pdf