UPC - GIA

Practice 1: Rule-Based Dialogue System

Voice and Dialogue Processing

Author:
Elena Alegret & Júlia Orteu

Setember 2023

1 Introduction

This practice focuses on creating a simple housing purchase assistant through a rule-based dialogue system.

Our objective is to develop a system that can help users find houses that match their preferences and needs.

This assistant uses predefined rules to identify and display available housing options. User preferences will be the basis for recommendations, and the system will guide the user through a series of questions to collect these preferences.

Therefore, it is based on creating a housing purchase assistant that helps users find the house of their dreams.

The structure of this document is divided into five fundamental sections:

- Introduction.
- Results.
- Conversation Flow.
- Integration into the Interface.
- Conclusions.

The *Introduction* provides an overview of our project, identifying the task we want to address and setting the objectives.

In the *Results*, the results of the exercises we have performed during the development of our practice are presented.

In the *Conversation Flow* section, a practical case of how our chatbot operates is presented. It provides a detailed diagram of how the system interacts with the user, collects information, and offers responses and recommendations. This part allows readers to understand how our assistant works visually.

To delve into the field of text assistants, the *Integration into Interface* section presents the modeling of a graphical chat interface to simulate how the housing assistant could appear in a real environment.

Finally, the *Conclusions* we have drawn from our practice are presented.

2 Results

To run the *House Buying Assistant* program, you need to call the main function using *Python3*:

- `$ python3 main.py`

2.1 Adding a New House Manually to the JSON File with ID 26

In this first point, a new house was added manually to the JSON file with the following characteristics:

- Id: 26
- Type: sale
- Bedrooms: 2
- Bathrooms: 2
- Price: 400k
- Square meters: 60
- Floor: 3
- Elevator: No
- Commercial use: Yes
- Terrace: No
- Location: Barcelona

Taking a first look at the JSON file allowed us to understand the data structure and see how questions and houses are stored.

2.2 Adding the Welcome Message Stored in the JSON File

This exercise is based on ensuring that every time the conversation with the user starts, the welcome message systematically appears on the chatbot's screen as a mandatory starting point.

```
Assistant: Welcome to the House Buying Assistant!
Assistant: How many bedrooms do you need?
Assistant: Enter your choice (1 - 5 bedrooms)
```

As you can see, the initialization meets the requirements.

2.3 Allowing the User to Quit the Program

The goal of this exercise is to allow the user to end the conversation with the chatbot using the word 'quit' or 'Quit'. When this word is detected, the conversation will be closed properly, and the user will exit the system.

```
Assistant: How many bedrooms do you need?
Assistant: Enter your choice (1 - 5 bedrooms)
User: quit
Assistant: Thank you for using the House Buying Assistant. Goodbye!
```

2.4 Completing the find_suitable_houses Function

The *find_suitable_houses* function displays to the user the suitable house(s) that meet their requirements.
Here is the function:

```python
def find_suitable_houses(data, user_preferences):

    """
    Finds suitable houses in the given data based on user preferences.
        :param data: A dictionary containing information about available houses.
        :param user_preferences: A dictionary representing the user's preferences.
        :return: A list of suitable houses that match the user's preferences.
    """

    suitable_houses = []  # Suitable houses for the user
    keys =  user_preferences.keys()

    for house in data["houses"]:  # For each house
        is_suitable = True

        for key in keys:
            if  (key in house and house[key] == user_preferences[key]) or \
                (key not in house) or \
                (key in ['floor', 'rent'] \
                      and int(house[key]) in user_preferences[key]):
                pass

            else:
                is_suitable = False
                break

        if is_suitable:
            suitable_houses.append(house)  # Suitable answer

    return suitable_houses
```

Firstly, it should be noted that the code has been developed based on the structure of the given JSON file.

The function is initialized by storing the keys of the *user_preferences* dictionary, which represent the preferences that have been accumulated based on the user's responses.

Next, a loop is initiated that allows iterating through each user preference and comparing it with the characteristics of the current house. In this way, if a house meets all the user's preferences, it is added to a list of suitable houses.

The filtering of houses is performed as follows:

- If the value of the house's feature matches the user's preferences, it is considered a valid match.
- If the feature does not exist in the house's information, it is considered a valid match.
  - *Clarification*: If the feature does not exist in the house, it indicates that the user has not chosen the offer type options (sale, rent). For convenience, rental properties do not have the 'price' label, and properties for sale do not have the 'rent' label.
- If the currently observed preference is *floor* or *rent*, and it matches the preferences, it is considered a valid match.

Therefore, if these conditions are met, the house is added to the list of suitable houses.

In the end, the result is the houses that are suitable for the user.

2.5 Adding the Option to Choose None for a Question

If the user decides not to answer the assistant's question or chooses an option that is not available in the database, the system, while notifying the user, proceeds to the next question without considering the omitted preference.

First case: The user does not respond to the question.

```
Assistant: Which city or neighborhood would you prefer?
Assistant: Options: L'Hospitalet de Llobregat, Esplugues de Llobregat, Santa Coloma
           de Gramenet, Barcelona
Assistant: Enter your choice:
User:
Assistant: I'm sorry, but the answer is not among the available options.
Assistant: What would be your minimum floor threshold?
```

Second case: The user chooses an option that is not available.

```
Assistant: Would you like to be in dispose of a terrace?
Assistant: Options: No, Yes
Assistant: Enter your choice (Yes - No)
User: I don't know.
Assistant: I'm sorry, but the answer is not among the available options.
Assistant: Elevator access: is that a preference for you?
```

2.6 Expanding the Chatbot

- **Type of Property**: Ask the client if they are interested in buying or renting the property, or if they prefer not to choose, with both options. It has been decided that this will be one of the last questions.

```
Assistant: Which type of housing would you prefer?
Assistant: Options: sale, rent
Assistant: Enter your choice (sale - rent):
```

- **Income**: If the client chooses the rental option, they will be asked about their monthly household income, as only properties below 35% of their income will be displayed.

```
Assistant: Which type of housing would you prefer?
Assistant: Options: rent, sale
Assistant: Enter your choice (sale - rent)
User:      rent
Assistant: As you've opted for renting, could you please provide
           your monthly household income?
User:      3000
Assistant: Thank you for sharing your household income.
Assistant: We will now display rental options that fit within your
           recommended maximum monthly rent of 1050.0
```

Furthermore, the system will not stop asking the user about their monthly household income until it obtains a valid response.

```
Assistant: As you've opted for renting, could you please provide
           your monthly household income?
User:      No
Assistant: Please provide a valid numerical value. Try again:
User:      A lot
Assistant: Please provide a valid numerical value. Try again: 3500
Assistant: Thank you for sharing your household income.
```

- **Floor Preference**: The user decides on the minimum floor level they prefer to live on. From this point onwards, only properties on the same level or higher will be displayed.

```
Assistant: What would be your minimum floor threshold?
Assistant: Enter your choice (0 - 8 floors)
```

- **Terrace Preference**: If the user is looking for properties with a terrace, only those with terraces will be displayed. If not, both options with and without terraces will be shown.

```
Assistant: Would you like to be in dispose of a terrace?
Assistant: Options: No, Yes
Assistant: Enter your choice (Yes - No)
```

- **Elevator Preference**: If the user prefers having an elevator, only properties with elevators will be displayed. Otherwise, options with and without elevators will be shown.

```
Assistant: Elevator access: is that a preference for you?
Assistant: Options: No, Yes
Assistant: Enter your choice (Yes - No)
```

- **Commercial Use Preference**: If the user is looking for a property for commercial use, only those properties that can offer this will be displayed. Otherwise, this property will not be considered in the choice of the property.

```
Assistant: It would be a commercial use habitat?
Assistant: Options: No, Yes
Assistant: Enter your choice (Yes - No)
```

3 Conversation Flow

In this section, we present the Conversation Flow of our assistant for buying/renting houses. The diagram shown in Figure 1 illustrates the logic of how our assistant operates, guiding users through the process of searching for and acquiring properties.

The conversation flow outlines the steps and interactions between the user and the virtual assistant, from the start of the conversation to the end, where the assistant presents the user with properties that match their preferences.

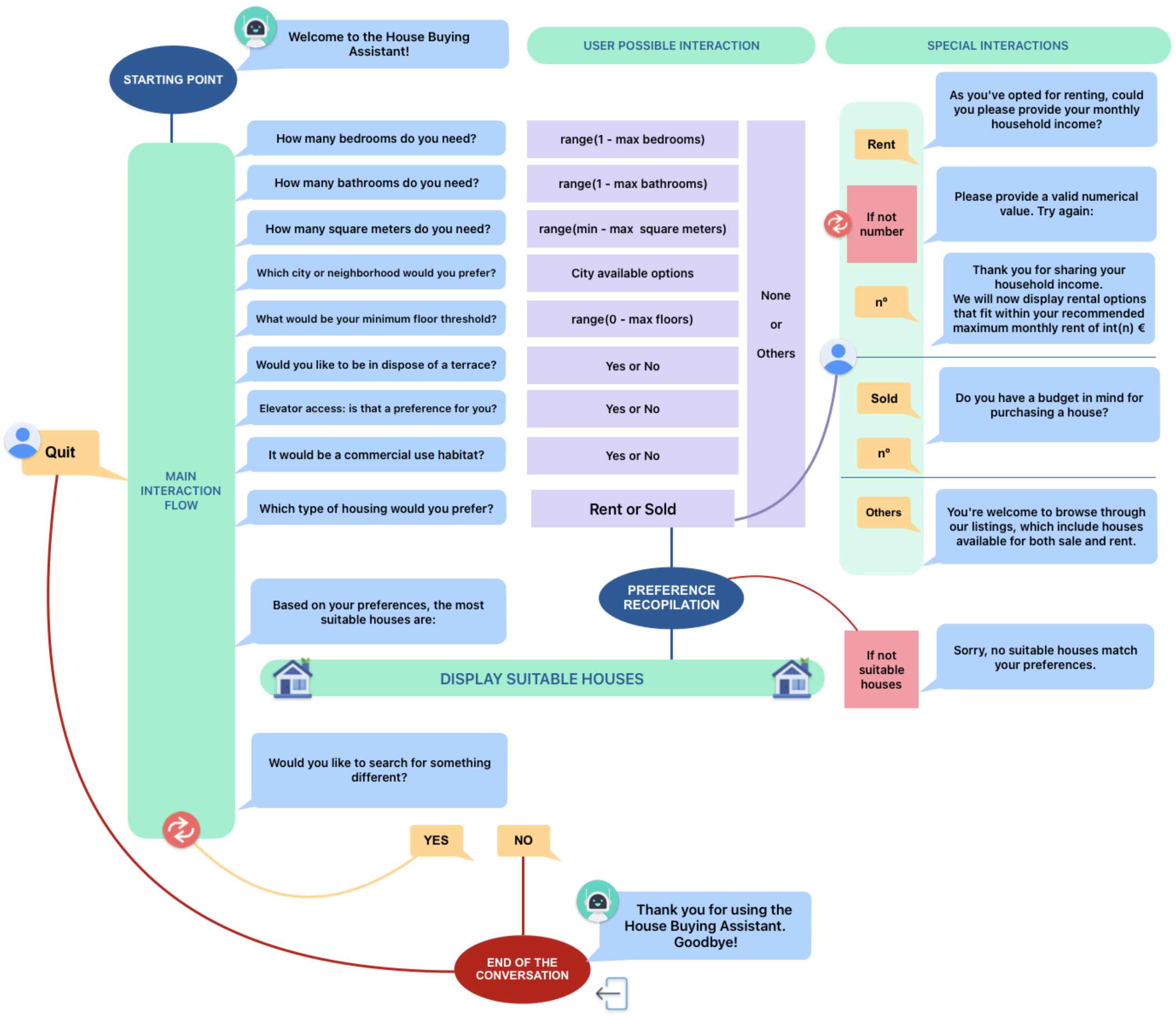
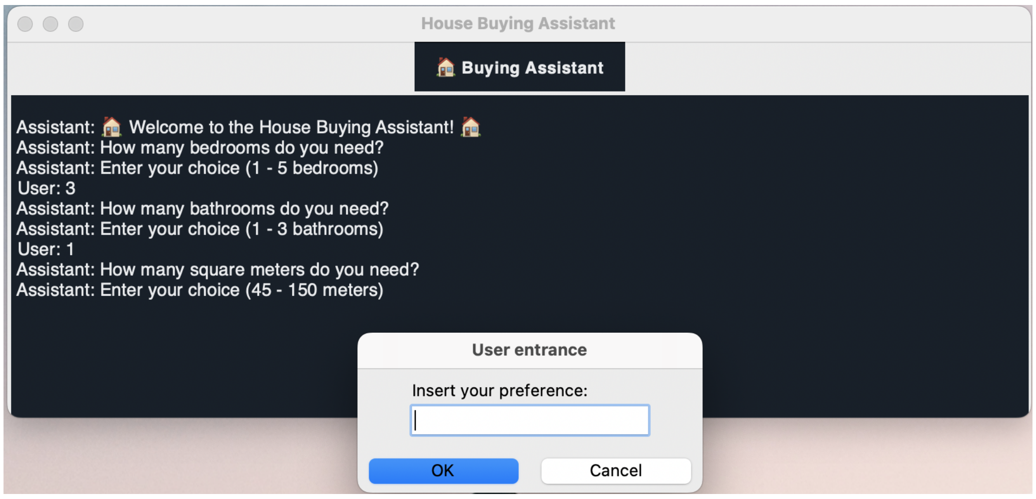

Welcome to the House Buying Assistant!
STARTING POINT
USER POSSIBLE INTERACTION
SPECIAL INTERACTIONS
How many bedrooms do you need?
range(1 - max bedrooms)
How many bathrooms do you need?
range(1 - max bathrooms)
How many square meters do you need?
range(min - max square meters)
Which city or neighborhood would you prefer?
City available options
What would be your minimum floor threshold?
range(0 - max floors)
Would you like to be in dispose of a terrace?
Yes or No
Elevator access: is that a preference for you?
Yes or No
It would be a commercial use habitat?
Yes or No
Which type of housing would you prefer?
Rent or Sold
None or Others
MAIN INTERACTION FLOW
Quit
Rent
As you've opted for renting, could you please provide your monthly household income?
If not number
Please provide a valid numerical value. Try again:
nº
Thank you for sharing your household income. We will now display rental options that fit within your recommended maximum monthly rent of int(n) €
Sold
Do you have a budget in mind for purchasing a house?
nº
Others
You're welcome to browse through our listings, which include houses available for both sale and rent.
PREFERENCE RECOPILATION
If not suitable houses
Sorry, no suitable houses match your preferences.
Based on your preferences, the most suitable houses are:
DISPLAY SUITABLE HOUSES
Would you like to search for something different?
YES
NO
Thank you for using the House Buying Assistant. Goodbye!
END OF THE CONVERSATION

Figura 1: Dialogue Flow Diagram

3.1 Integration into Interface

Building upon the logic of the real estate assistant implemented in the previous sections, the code has been adapted to model a graphical interface that simulates a chat.



```
Assistant: 🏠 Welcome to the House Buying Assistant! 🏠
Assistant: How many bedrooms do you need?
Assistant: Enter your choice (1 - 5 bedrooms)
User: 3
Assistant: How many bathrooms do you need?
Assistant: Enter your choice (1 - 3 bathrooms)
User: 1
Assistant: How many square meters do you need?
Assistant: Enter your choice (45 - 150 meters)
```

Figura 2: Screenshot of the graphical chat interface.

In this section, the aim was to extend the knowledge gained during the creation of the assistant to visualize it in a more realistic environment. Furthermore, a more natural and user-friendly interaction with the user has been pursued.

As a tool, the Python package *Tkinter* has been used. Tkinter is a binding of the Tcl/Tk graphical library.

To run the program, you need to call the modified main function using Python3:

- `$ python3 main_interface.py`

4 Conclusions

During the development of the real estate recommendation chatbot project, we have learned many valuable skills and knowledge that can be applied in the professional world. We had the opportunity to work on a real project that combines programming and user interaction, enhancing our software development capabilities. We learned to work with Natural Language Processing (NLP) and utilize libraries like NLTK for text preprocessing and user communication.

Furthermore, we gained experience in structuring and organizing a programming project effectively. We learned how to design a user interface with Tkinter, improving our user interface and design skills. We also acquired skills in data management and storage in JSON format, as well as accessing and utilizing data in our project. We developed data processing functions and algorithms to identify and display properties that match user preferences.

It's worth noting that each member of the group works differently, and this diversity allowed us to approach this project from two different perspectives. Working in a group has been a key tool, as it allowed us to progress thanks to each other's ideas and overcome any obstacles we encountered.

In conclusion, this project has been an opportunity to develop our skills in programming virtual assistants guided by user interactions.