
Supervised and Experimental Learning

User Manual

CBR System for a Synthetic Task: Cocktail Creator

Professor

SANCHEZ MARRE, Miquel

Team members

BEJARANO SEPULVEDA, Edison Jair
edison.bejarano@estudiantat.upc.edu

REYES FERNANDEZ, Grecia
greCIA.carolina.reyes@estudiantat.upc.edu

DEL REY JUÁREZ, Santiago
santiago.del.rey@estudiantat.upc.edu

ZURITA MARTEL, Yazmina
yazmina.zurita@estudiantat.upc.edu

Universitat Politècnica de Catalunya

C/Jordi Girona Salgado,1-3 08034 BARCELONA Spain

Master's degree in Artificial Intelligence

Barcelona, 2022

Contents

1	Description of the system goals	2
2	Prerequisites	2
3	Graphical User Interface	2
3.1	Introducing your preferences	2
3.2	Sending the request	3
4	Comand Line Interface	3
5	System functionalities	6

1 Description of the system goals

The Cocktail Creator CBR system is a cocktail recommendation system able to suggest a cocktail recipe based on the constraints described by the user in the graphical user interface (GUI) or command line. These constraints can be preferences about the drink type, glass type, type of alcohol, etc. Additionally, the user can specify ingredients to exclude from the recipe.

The system has the following phases:

- Retrieve the most similar cases to the new case from the case library, at least 5 cases, depending on the constraints entered by the user.
- Adapt information from retrieved cases to fit the new case.
- Evaluate from the proposed solution, how well the adapted solution addresses the current problem (new case).
- Learn the new problem and solution as a case, to keep them in the case library for possible future problem solving. Successful and unsuccessful solutions are learned.

2 Prerequisites

In order to be able to run the system without problems, you need to ensure your Python environment has the appropriate dependencies installed, and that the data is located in the correct folders.

To install the dependencies, you have at your disposal a requirements.txt file in the root directory of this project.

Also, you need to make sure that there is a folder named data in the project folder, and that it contains the Case Library file named case_library.xml.

3 Graphical User Interface

This section explains how to run and use the system from the GUI.

To start-up the GUI, you only need to run the app.py file under src/app. Then you will be greeted with a single window GUI.

3.1 Introducing your preferences

The use of the GUI is straight forward. You have available several preferences regarding the recipe that you can fill in. The application only allows you to select a single type of drink and glass for the request.

To include multiple types of alcohols and basic tastes to the recipe you have an input text line. Once you start typing the system will prompt you with a list of available options matching your input text. To add the entry just press Enter. In case you want to remove one of the selections, you only need to double click the desired item.

In the same fashion, you can input the ingredients that you want to include or exclude from your cocktail. The procedure to remove elements from the lists is the same as described above. In this case, to add the elements to either of the two lists you must explicitly click on the appropriate button or use the button's shortcut which is Alt+key. Being key the one corresponding to the underlined letter of the button's name.

One thing to keep in mind is that the system will prompt you with an error if you try to add an alcohol, taste or ingredient that does not exist in the system's database.

3.2 Sending the request

After you have filled your preferences, you can press the Run button to send your request to the system. If you have not introduced a name for your recipe the system will warn you and ask you to do so. Otherwise, you will be shown the recipe the system used to build your cocktail and the adapted recipe which should match your preferences. After this, you are expected to evaluate the proposed solution with a score between 0 and 10. Being 0 for very unsatisfactory results and 10 for very satisfactory results. After you send the evaluation, you can introduce new preferences and start again.

4 Comand Line Interface

A second approach to interact with the system is through a command-line interface (CLI). To star-up the CLI, run the cbr-cli.py file under src/app. You will receive a welcome message and a petition to enter your preferences for the drink.

First, you must introduce a name for the recipe. If this field is left blank, an error message will be printed and the system will request you to enter a name again. Once you have given a name to the recipe, press enter to continue filling the remaining fields.

In the next fields, you will be asked for your preferences regarding the recipe. The values entered must be present in the system's database or you will receive an error message. If you write "suggest" and press enter, the system will give you a list of examples of values you can enter.

Following the name of the recipe, you must input the category of the drink and the type of glass. Only one value is admitted in these fields. The next fields are the type of alcohols, tastes and the ingredients you want in the recipe and the ingredients you want to exclude from it. You can enter an unlimited number of values in these fields or leave them blank. Note that you can not enter an ingredient in the exclusion list that is already in the inclusion list.

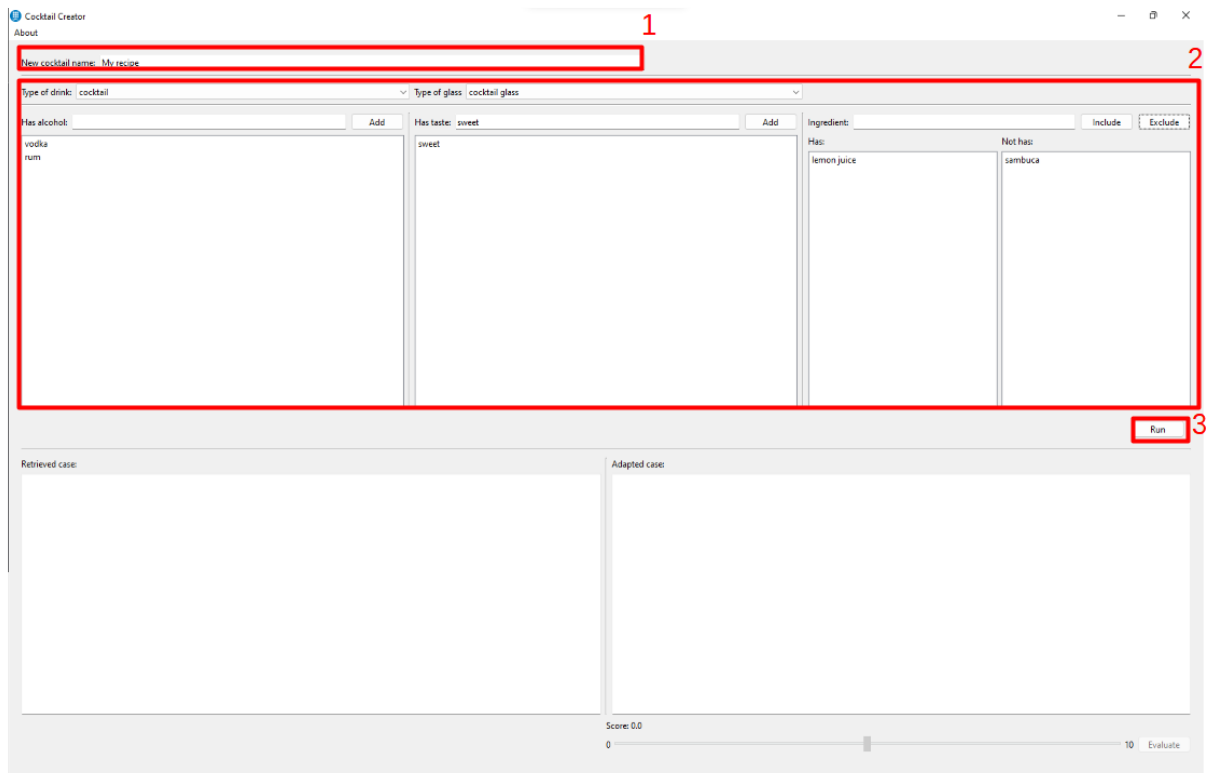


Figure 1 – Steps to fill in the request preferences.

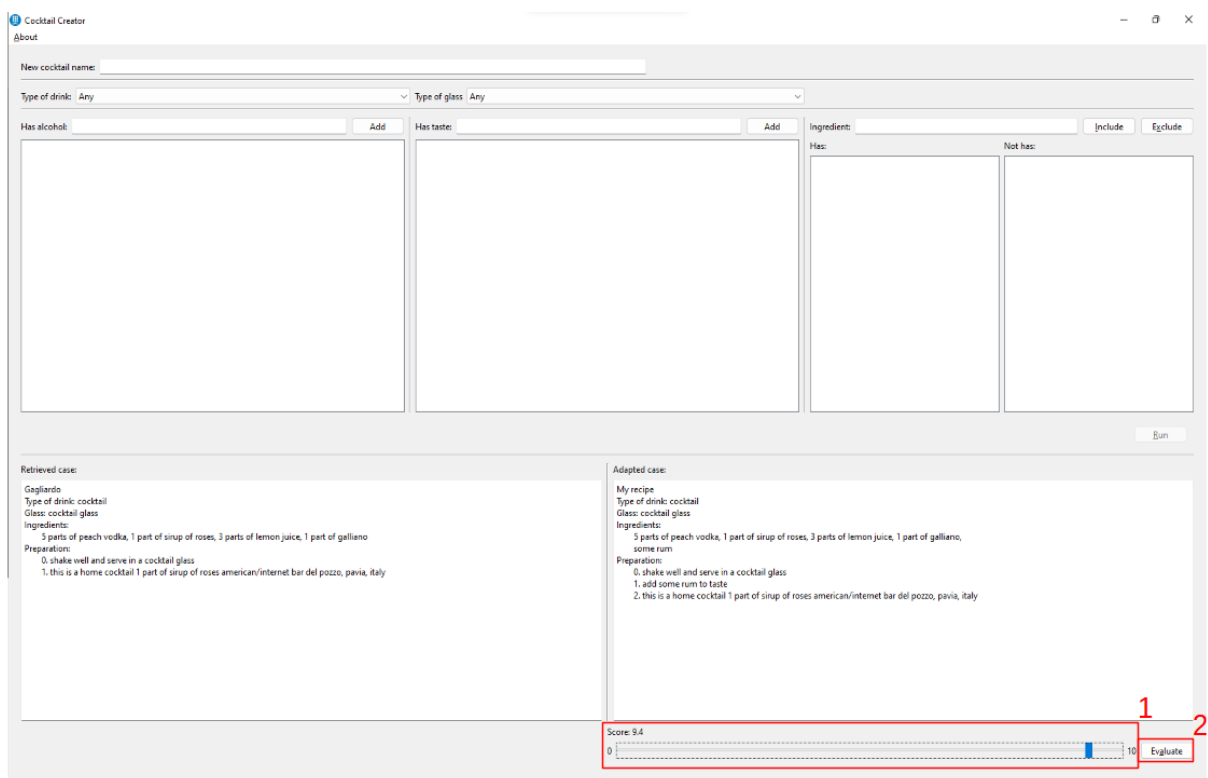


Figure 2 – Example of a result and how to evaluate it.

Lastly, the application will show you the retrieved recipe from the system's database and the adapted recipe. The system will ask you to evaluate the adapted recipe with a value from 0 to 10. Once you have introduced the value and pressed enter, you will see a closing message and the application will close.

```
- Welcome to CBR Cocktails.
- Please, enter the name of the recipe and your preferences. Write "suggest" to see examples.

- Name of the recipe: MyRecipe

- Type of drink (e.g.: beer, ordinary drink): coffee
-- Error. "coffee" is not in the library. Another value must be specified.
- Type of drink (e.g.: beer, ordinary drink): suggest
-- Suggestions: coffee / tea, shot, punch / party drink, cocktail, milk / float / shake
- Type of drink (e.g.: beer, ordinary drink): coffee / tea

- Type of glass (e.g.: old-fashioned glass, pint glass):
-- Error. A value must be specified.
- Type of glass (e.g.: old-fashioned glass, pint glass): coffee mug

- Type of alcohol (e.g.: gin, triple sec):

- Taste of the drink (e.g.: sour, salty): suggest
-- Suggestions: salty, cream, water, egg, bitter
- Taste of the drink (e.g.: sour, salty): bitter, cream

- Ingredients (e.g: cherry, rum): rum, milk, sugar, coffee

- Ingredients to exclude (e.g: banana, vodka): milk
-- Error. Ingredients to exclude can not be present in the ingredients to include list.
- Ingredients to exclude (e.g: banana, vodka): whipped-cream
```

Figure 3 – Example of use of the CLI (1/3). Input of the query.

```

- Here is the retrieved recipe:
Cafe Savoy
Type of drink: coffee / tea
Glass: coffee mug
Ingredients:
    1/2 oz of triple sec,  of coffee, 1/2 oz of milk,  of brandy
Preparation:
    0. fill mug almost to top with  of coffee.add 1/2 oz of milk, 1/2 oz of triple sec and  of brandy
    1. stir

- Here is the adapted recipe:
MyRecipe
Type of drink: coffee / tea
Glass: coffee mug
Ingredients:
    1/2 oz of triple sec,  of coffee, 1/2 oz of milk,  of brandy
    some rum, some sugar
Preparation:
    0. fill mug almost to top with  of coffee.add 1/2 oz of milk, 1/2 oz of triple sec and  of brandy
    1. add some sugar to taste
    2. add some rum to taste
    3. stir

```

Figure 4 – Example of use of the CLI (2/3). Result generated.

```

- Evaluate this recipe with a score from 1 to 10 (e.g.: 7.5): -1
-- Error. The score must be between 0 and 10.
- Evaluate this recipe with a score from 1 to 10 (e.g.: 7.5): 11
-- Error. The score must be between 0 and 10.
- Evaluate this recipe with a score from 1 to 10 (e.g.: 7.5): one
-- Error. The score must be a float.
- Evaluate this recipe with a score from 1 to 10 (e.g.: 7.5): 8

- Evaluation sent.
- Done.

```

Figure 5 – Example of use of the CLI (3/3). Evaluation.

5 System functionalities

The different functionalities of the interface have as main purpose to deliver to the user the possibility to use the system in an easy way. To this aim, the system is composed of three basic functionalities:

- **Recipe name:** The system expects the user to introduced a name for the new recipe that will be generated from the selected constraints.
- **Inclusion constraints:** The system enables the user with the possibility of selecting different parameters that the adapted case must include. These parameters are the drink type, glass type, alcohol types, basic taste types, and ingredients.

- **Exclusion constraints:** The system allows the user to specify a set of ingredients that must be excluded from the adapted case.
- **Assorted restrictions:** For this specific variable constraint it was being considered, since in the system the only variable that the user introduces in a personalized way is the name of the new adapted cocktail, where it is allowed to be provided by the user.
- **Evaluation score:** When the adaptation phase is completed, the system asks for the user to input an evaluation score representing its satisfaction score with the given solution.

For the drink and glass type only one value can be selected at a time. In the case of alcohol type and basic taste, the user can select as many options as desired. The selection of the ingredients work in a similar way with the addition that once an ingredient is selected to be included or excluded it can not be added to the opposite list.