LBA

Minerva University

CS152 - Harnessing Artificial Intelligence Algorithms

December 11, 2023

1. Details of Expert System

The knowledge base contains detailed information about cafes in Buenos Aires, including locations, drink preferences, dietary restrictions, minimum rating, wifi availability, and other attributes that help discern the optimal cafe for users based on user preferences. Specifically, the system will interact with users through the following askables:

<table>
<tr>
<td>

- Which location do you prefer?
  - Palermo
  - Puerto Madero
  - San Martin
  - San Telmo
  - Recoleta
- What are you craving?
  - Coffee
  - Snacks
  - Meal
- Any dietary restrictions?
  - Gluten Free
  - Vegetarian
  - Vegan
  - Dietary restrictions

</td>
<td>

- Minimum rating of cafe?
  - 0-5
- Does it have Wifi?
  - Yes
  - No
- How expensive of a cafe do you prefer? (Cup of coffee)
  - $500 ~ $1000
  - $1500 ~ $2000
  - $2000 ~ $3000
- Is the cafe a Bares Notables?
  - Yes
  - No

</td>
</tr>
<tr>
<td>

- What type of cafe do you prefer?
  - Study Cafe
  - Tango Cafe
  - Animal Cafe
  - Eateries

</td>
<td>

- Payment Option
  - (cash) Dollars
  - (cash) Pesos
  - MercadoPago
  - Debito
  - Credito

</td>
</tr>
</table>

Currently, many Minerva students frequently ask for cafe recommendations in the group chat. However, oftentimes, they fail to find a satisfactory cafe, if they involve several constraints such as being vegan, location, or price levels. Given this initial state, the goal state would include facilitating the process of cafe searches, based on the needs individual students require. In an ideal goal state upon implementing the expert system, we envision students being able to utilize the tool for efficiently exploring the city with minimal research; we included askables regarding Bares Notables, a cafe designated as the government that represents the culture of Argentina, to

also aid Minerva students who love experiencing a new culture. The scale of the applicability is not only limited to Minervan students, but can also serve as an expert system for locals who lack time to individually conduct research on which cafes to visit.

1. **Data Collection**

The data regarding each cafe was collected through different methods to ensure accuracy and reliability. Given Argentina's volatile price levels, information on the internet is often outdated. To ensure currency of the data, the price level of the cafe was investigated through surveying students in M25's group chat who had been to that particular cafe, in the previous 2 weeks. For cafes that none of the M25s have visited, we have dialed the cafe to survey the prices. For the askable regarding Bares Notables, we have consulted the official Argentine Government website (n.d.) for accuracy. The rest of the data regarding Askables were available on Google Maps, in the 'About' tab. We evaluated the reviews to ensure that cafes were not overstating their features.

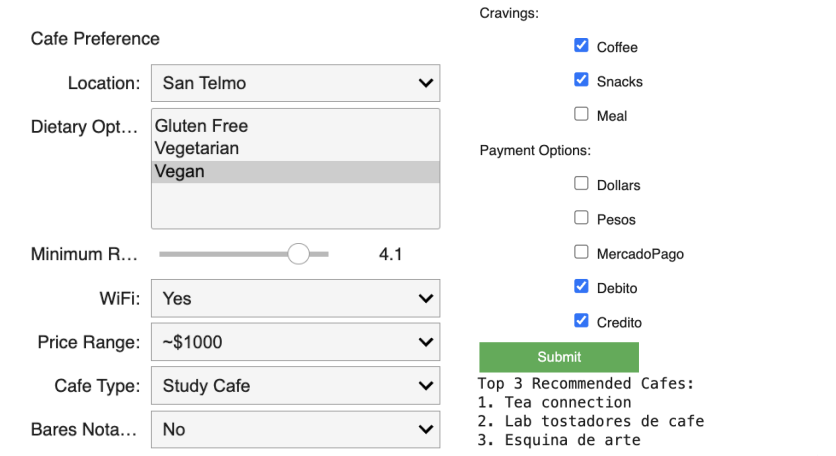
2. **Logic of Expert System (Fig.1, available in Appendix D)**

Within our Expert System, we have our knowledge base of cafes. The user will input their preferences regarding each askable, and every time it matches the cafe type, we will increment the Cafe's ranking by 1. For example, If the user's preference regarding location is Palermo and the cafe is located in Palermo, we do Ranking +=1. In the end, we will output the top 3 cafes, based on the ranking values.

4. **Expert System**

The expert system is available in the attached file and in Appendix C. Also, it is available in https://colab.research.google.com/drive/1c8P1SHqPeDvvbD4DK2ryWiuMLlx_gTiv?usp=sharing.

5. Test Cases



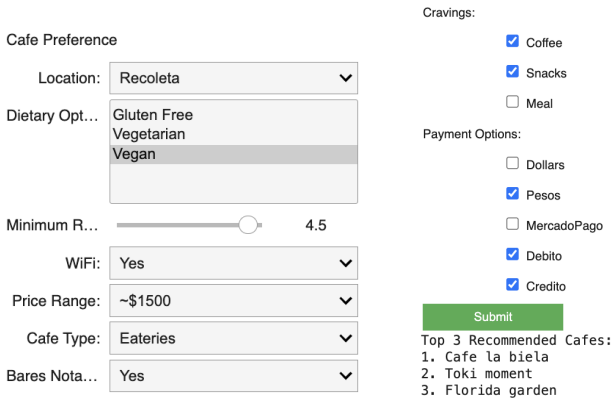

```
prolog.assertz("location('tea_connection', san_telmo)")
prolog.assertz("cravings('tea_connection', [coffee, snacks])")
prolog.assertz("dietaryOptions('tea_connection', [vegan, gluten_free])")
prolog.assertz("wifi('tea_connection', yes)")
prolog.assertz("cafeType('tea_connection', study_cafe)")
prolog.assertz("baresNotables('tea_connection', no)")
prolog.assertz("paymentOptions('tea_connection', [debito, credito])")
prolog.assertz("rating('tea_connection', 4.1)")
prolog.assertz("priceRange('tea_connection', 1000)")
```

In this test case, we check if the queries regarding attributes with spacing and capital letters, like San Telmo, are translated correctly to query san_telmo in the KB. Indeed, we observe that Tea connection Cafe, which satisfies all conditions, is outputted as top-ranked. Additionally, the outputs are correctly formatted to print Tea connection instead of tea_connection.

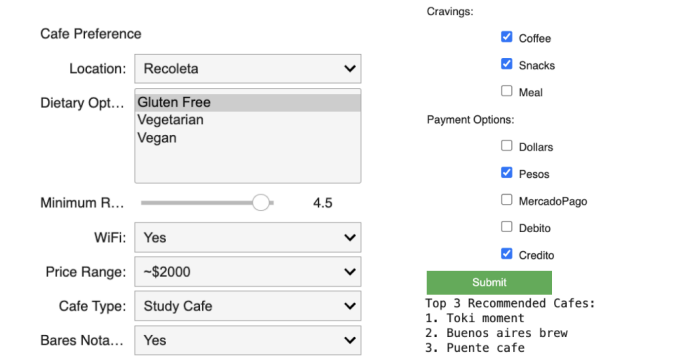
```
("location('cafe_la_biela', recoleta)")
("cravings('cafe_la_biela', [coffee, snacks])")
("dietaryOptions('cafe_la_biela', [vegan, gluten_free])")
("wifi('cafe_la_biela', yes)")
("cafeType('cafe_la_biela', eateries)")
("baresNotables('cafe_la_biela', yes)")
("paymentOptions('cafe_la_biela', [pesos, debito, credito])")
("rating('cafe_la_biela', 4.5)")
("priceRange('cafe_la_biela', 1500)")
```

```
("location('toki_moment', recoleta)")
("cravings('toki_moment', [coffee, snacks])")
("dietaryOptions('toki_moment', [gluten_free])")
("wifi('toki_moment', yes)")
("cafeType('toki_moment', study_cafe)")
("baresNotables('toki_moment', yes)")
("paymentOptions('toki_moment', [pesos, credito])")
("rating('toki_moment', 4.9)")
("priceRange('toki_moment', 2000)")
```

```
("location('florida_garden', recoleta)")
("cravings('florida_garden', [coffee, snacks])")
("dietaryOptions('florida_garden', [])")
("wifi('florida_garden', yes)")
("cafeType('florida_garden', eateries)")
("baresNotables('florida_garden', yes)")
("paymentOptions('florida_garden', [pesos, mercado_pago])")
("rating('florida_garden', 4.3)")
("priceRange('florida_garden', 3000)")
```



In this test case, we observe if the ranking utility calculations are done correctly. For the first rank, we have Cafe la biela, satisfying all the conditions, with 12 categories satisfied. Toki moment satisfies 8 categories, while Florida garden satisfies 7 categories.




```
prolog.assertz("location('toki_moment', recoleta)")
prolog.assertz("cravings('toki_moment', [coffee, snacks])")
prolog.assertz("dietaryOptions('toki_moment', [gluten_free])")
prolog.assertz("wifi('toki_moment', yes)")
prolog.assertz("cafeType('toki_moment', study_cafe)")
prolog.assertz("baresNotables('toki_moment', yes)")
prolog.assertz("paymentOptions('toki_moment', [pesos, credito])")
prolog.assertz("rating('toki_moment', 4.9)")
prolog.assertz("priceRange('toki_moment', 2000)")
```

For our LBA component, we queried for a cafe that specifically suits our needs. We searched for a Bares Notables study cafe that has wifi, near Recoleta, where we were located before going to the cafe. Regarding food, we wanted to get snacks and coffee, and Yubin was a gluten-free consumer. For payment, we wanted a mid-range price cafe with ~2000 pesos and wanted to pay in either pesos or credit cards. Upon our query, we found that the Toki Moment cafe in our KB indeed satisfies our needs.

6. Visiting Location

As outputted in our last test case, we went to the Toki Moment cafe (Appendix E). As Yubin is a gluten-free consumer, she was able to enjoy the snacks there, as per our needs.

Bibliography

El Gobierno de Argentina. (n.d.). *Bares notables*. Buenos Aires - Cultura - Patrimonio de la Ciudad. https://buenosaires.gob.ar/cultura/patrimonio-de-la-ciudad/acciones/bares-notables

Google. (n.d.). Google maps. https://www.google.com/maps

Appendices

Appendix A. Contributions

Saad Asad: Developed the visualization component for the knowledge base. Designed and implemented the user interface form. Created encapsulating functions to transform form data into Prolog assertions. Debugged the querying system. Established the PySWIP setup.

Len Park: Wrote the report and HC/LO index. Collaborated with Fady to format the Knowledge Base in Prolog and translate it to PySWIP. Streamlined the KB for consistency, and devised test cases.

Fady Hanna: Collected the KB data, formatted it in prolog syntax, and tested functionality by querying the KB then wrote it as pyswip.

Yubin Park: Created the utility calculation function to query the top 3 cafes based on user preferences inputted. Streamlined the code to create a seamless connection between Saad's user interface and the utility function.