CS152 Location-Based Assignment

Adrian Goedeckemeyer, Guilherme Nazareth, and Roujia Wen

Surveyed Restaurants

In this location-based project, we surveyed 30 restaurants in total (see Table 1).

일일향 (日日香)	대우식당	유타로 (雄太郎)
김돈이	배꼽집	이야기 하나
마루심 (〇心)	Cuisson 82	Brick Oven New York Pizzeria
장꼬방 김치찌개	Brown Bread	봉산옥
Jeremy's Burger	남서울민물장어	Do Chef Napoli Pizzeria
금계찜닭	Cafe 413 Project	부산양곱창
교대 이층집	진대감	Alla Prima
풍년집	윈윈수산	Retrooven
스시몬 (寿司文)	재패니즈 다이닝 안심 (安心)	Da Pitta
바베쿡스	이치류 (一流)	용무있습니까

Table 1 - A list of restaurants surveyed and included in the expert system.

Askables

Questions	**Choices**
Do you want take-out?[1]	"Yes", "No"
What type of food do you want?[2]	"Chinese", "BBQ", "Japanese", "Korean", "American", "Western", "Italian", "Asian"
What price range is acceptable?[2]	"$", "$$", "$$$"
Must have English menu?[1]	"Yes", "No"

Do you have dietary restrictions?[2]	"Vegan", "Vegetarian", "Gluten Free", "Halal"
Within a walking distance of (km)[3]	A float $\in [0, 10]$
Do you want a place that opens late?[1]	"Yes", "No"

Table 2 - A list of askables and range of choices in the expert system.

Type 1 questions (marked with superscript) are binary questions. In the GUI there is only one checkbox marked “Yes” for each of these questions, which can be unchecked. This design makes it convenient for the users, since if they don’t care about a specific criteria, they can simply leave the question unanswered, which means that no restaurants will be filtered out on the basis of this question. Type 2 questions are multiple choice questions. Selecting zero options for such a question also indicates that the user is open to all possibilities under this criteria. Type 3 question accepts a numerical value as input. Leaving it blank indicates that the user does not care about this filter.

In this project, the questions are presented in a GUI using TkInter, and answers are then collected, converted and passed onto a Prolog query system through PySWIP.

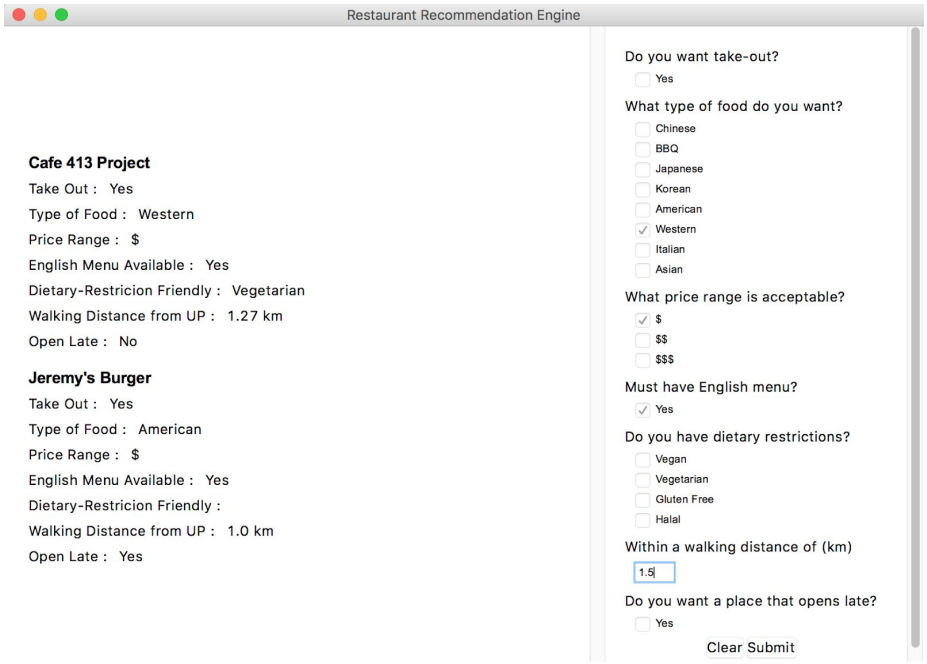
Prolog Rules

Each preference encoded in the responses is asserted into the prolog KB using a “known/3” predicate. This allows us to remove all preferences easily before the next query. The prolog KB also contains some rules regarding the hierarchy of food-types: if “Western” is selected, the types “American”, “Italian” and “Mexican” are acceptable as well, similar for “Asian”. Each restaurant suggestion in prolog consists of the facts that are required to understand

the restaurant, such as type of food, price and distance from Urban Place and if existing the restrictions that apply, such as no take-out, no english menu or not open late at night as well as dietary restrictions that are not accommodated. If a query contained dietary restrictions or requires a restaurant to be open late at night, these statements will fail, thus excluding such restaurant, while restaurants without that specific restaurant will continue to be shown. The conditions for restaurants in the KB are first checked for type and then price as these have the potential for directly excluding the most restaurants, thus saving us the work of checking all other restrictions.

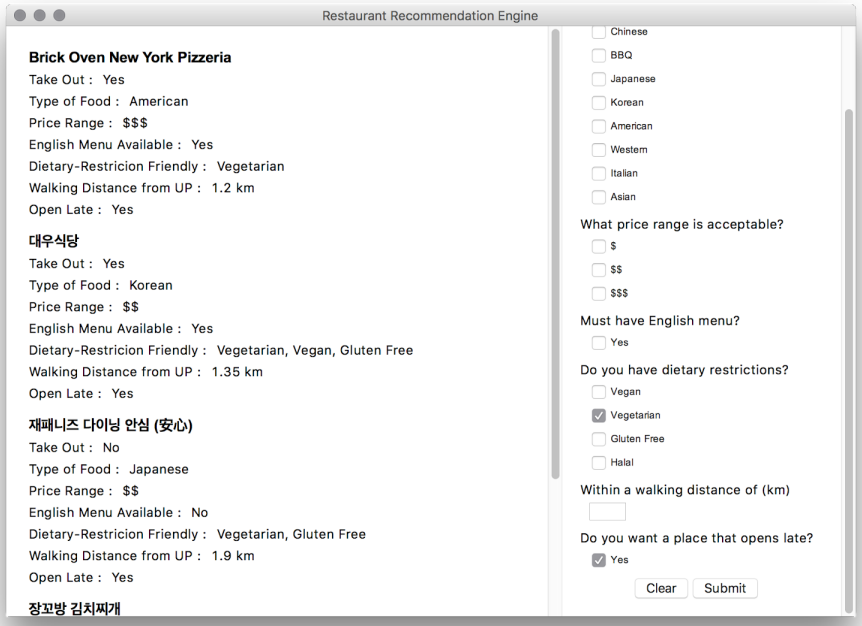
Major Test Case

Querying:

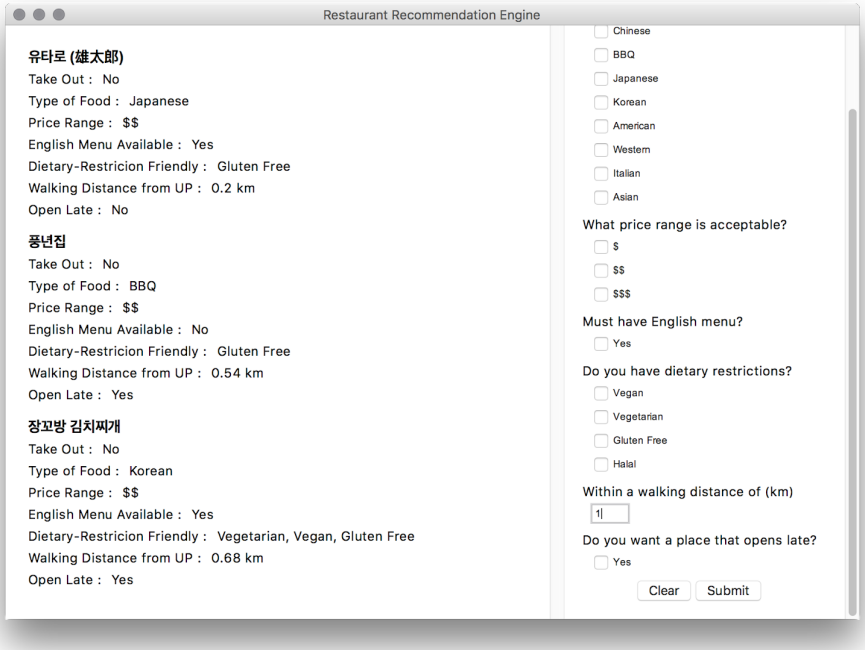


Other Test Cases

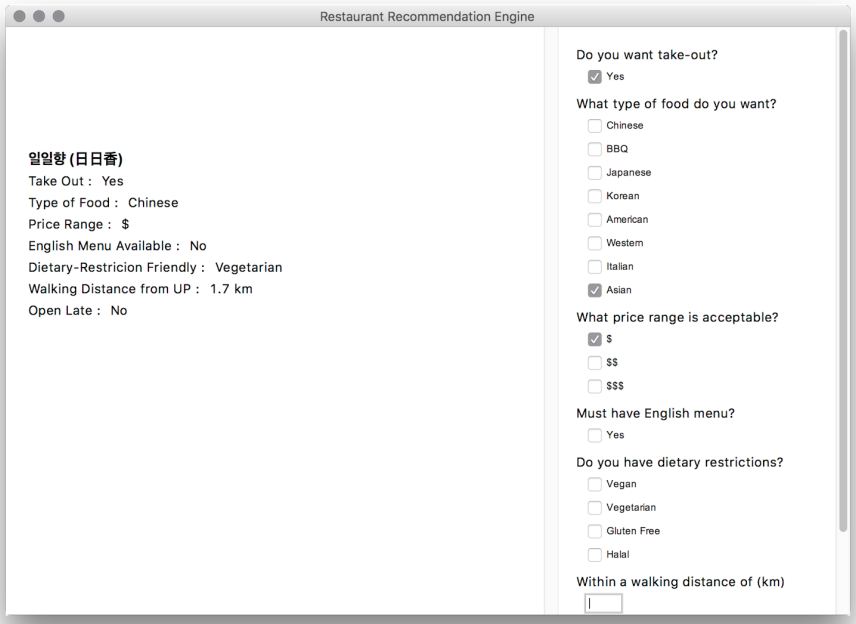
I'm vegetarian and it's midnight



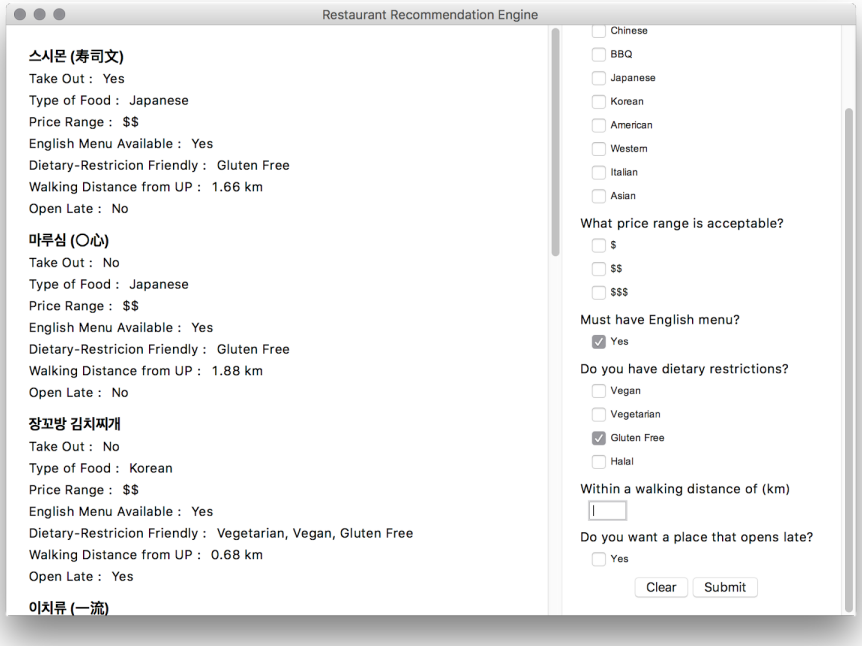
I'm lazy and don't want to walk



I would like a cheap Asian take-out meal.



I am allergic to gluten and I don't read Korean.



Individual Contributions

Roujia

Roujia was responsible for creating the GUI using Python and TkInter, standardizing I/O and intermediate data format, final integration and refinement of the code package. She wrote the "Surveyed Restaurants" and "Askables" sections of the report.

Guilherme

Guilherme helped with formulating the format of the askables, and was responsible for collecting the data by scraping Foursquare in areas nearby the Urban Place Gangnam Hotel, cleaning up the data and further researching for categories not present in Foursquare, and compiling the results in a CSV file and in the Prolog KB.

Adrian

Was responsible for finalizing the rules and logic to be used inside prolog and figuring out the best format of representation of restaurants inside prolog to be used for the restaurants surveyed for our purposes as well as writing the function that connects a query in python to the prolog KB by making assertions. Wrote the "Prolog Rules" section in this paper.

References

Merritt, D. (2001). Building Expert Systems in Prolog. *Amazi. inc*, *5861*. Retrieved from http://www.amzi.com/ExpertSystemsInProlog/xsipfrtop.htm