Commit: Final Project Checkpoint

Author: Yiu-Wen Yu

Date: 2018/04/11

After some trial and error, I revised the final project as below description.

**1. Program Description**

1. Upon initiation, the program displays an interactive command, which asks user to type in the food/restaurant in interest.
   1. Interactive command includes (ideally):
      * Help instruction <help>: display the description of the program
      * Search restaurants <search keyword>: search restaurants based on the keyword
      * Check past history <history>Note 1 : display the search history of the user
      * Exit the program <exit>
2. Based on the input, the program fetches data from Yelp using its API, and return a list of 10 restaurants based on the average rating.
3. At the same time, the program stores the search keyword in the database. The data in this database is used to display “trending search”.
4. After the program prints out the list of 10 restaurants, another interactive command will be displayed so that user can choose his/her next action.
   1. Interactive command includes (ideally):
      * Show map <map>: display the 10 restaurants on the map (see step 5)
      * See reviews of a restaurant <review restaurant>: display 10 most recent reviews of the selected restaurant (see step 6)
      * Check Open Table <opentable>: display the list of restaurants that user can make reservation via Open Table (see step 7)
      * New search <search keyword>: search restaurants based on the keyword
      * Help instruction <help>: display the description of the program
      * Check past history <history>: display the search history of the user
      * Exit the program <exit>
5. Using the location data retrieved from Yelp API, the program generates a new window displaying the map of 10 restaurants.
6. Based on the name of restaurant that user types in, the program accesses Yelp API “/business/{id}/reviews” to get the most recent reviews of that restaurant. Then prints out the reviews using Flask.
7. The program takes the list of restaurant as input, access the Open Table API to search whether the restaurant is using Open Table. The program prints out only restaurants that are available on Open Table.

**2. Data Source**

I am still using Yelp and Open Table. The only change in this part is that I will use both “business search” and “review search” on Yelp.

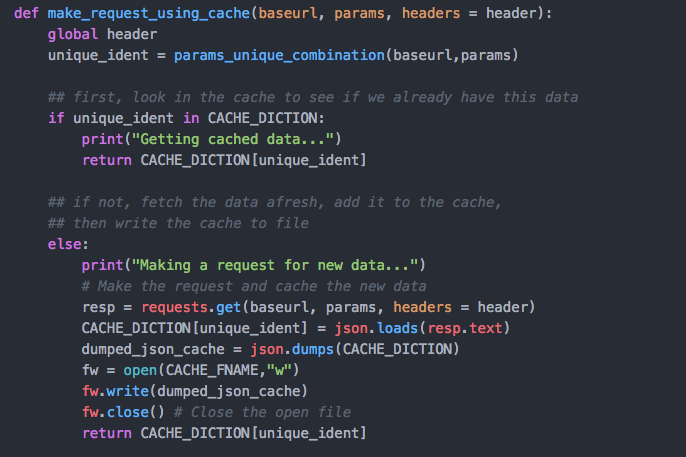
**3. Presentation options & tools**

1. Flask: Display the most recent reviews of a restaurant (step 6)
2. Plotly: Display the map of searched restaurants (step 5)
3. Interactive command \*2: Initial prompt (step 1) and the second prompt (step 4).

**4. Progress**

1. Data fetching:
   1. Yelp Fusion API:
      * Successfully retrieved data from “/business/search” API
      * Successfully access API using caching (code below)

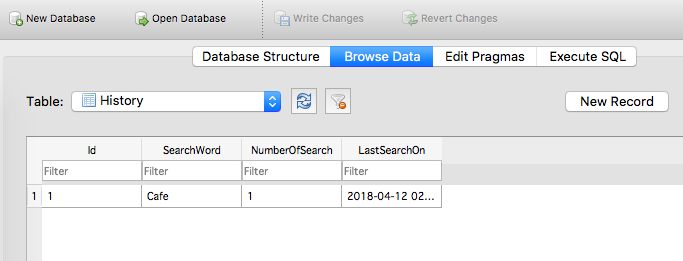






* + - To-do: retrieve data from “/business/{id}/reviews”
  1. Open Table API:
     + Haven’t started

1. Data storing:
   1. Save search history: In progress
      * Successfully store search keyword:



* + - Haven’t combined the save function into interactive command! (I used a separate code to test whether the save function works)

1. Presentation tools:
   1. Flask: Haven’t started
   2. Plotly: Haven’t started
   3. Interactive command:
      * Initial command (step 1): Finished
      * Follow-up command (step 4): In progress (60% completed)

**Note 1:**

Originally, I wanted to do “check what search term is trending in the past month”. The idea was to store all the search keywords from all users, and return the most frequent search keyword upon request. However, I decided to change this function to check the past search history for a user, because I don’t think my program can gather data from different users. Since the database created by the program is on the local device (e.g., the user’s desktop), there is no data entry from other users that I can use to aggregate.

**Github Link:**

You can find my final project codes here!

Commit: GSI Feedback

Author: Jie-Wei Wu

Date: 2018/03/29

1.

You didn't mention how you gonna store the data.

One thing I will suggest is to list each table and the columns of them (like hw 11) to help you have a clear picture of each table. Also make sure you will have two tables when implementing. This is the requirement of the final project.

2.

Also think about the user scenarios. What prompts/commands and what’s the display? That will make it easier for you to design the interaction interface as well as the logics of parsing the command.

Commit: Original Project Proposal

Author: Yiu-Wen Yu

Date: 2018/03/25

**SI 507 – Final project proposal**

Section: 009

GSI: Jie-Wei Wu

Student: Yiu-Wen (Evelyn) Yu

UMID: 88668251

**1. Program Description**

By entering the type of cuisine (e.g., Asian food or American food), users can get a list of 10 restaurants that have the highest customer rating from Yelp. After seeing the list, users can then check whether they can reserve tables via OpenTable.

**2. Data Source**

1) Yelp

* Planned usage: I plan to use the “/business/search” API or Yelp Fusion.
* Source URL: <https://api.yelp.com/v3/businesses/search>
* Parameters: “categories”
* Documentation: <https://www.yelp.com/developers/documentation/v3/business_search>
* Challenge score: 4 (Web API you haven’t used before that requires API key or HTTP Basic authorization)

2) OpenTable

* Planned usage: Based on user’s input, search whether user can reserve a table at the selected restaurant on OpenTable
* Source URL (Need to do some more research on API and OAuth): <https://oauth.opentable.com/api/v2/oauth/token?grant_type=client_credentials>
* Documentation: <https://platform.opentable.com/documentation/>
* Challenge score: 6 (Web API you haven’t used before that requires OAuth)

**3. Presentation options & tools**

1) A map that shows 10 restaurants via **Plotly**.

2) An interactive command line that asks users which restaurant user to check on the OpenTable.

3) A bar chart shows the rating of 10 restaurants via **Plotly**.

4) A line chart shows the change in average rating of the selected restaurant in the past 4 weeks.