**Wesley Upham**

**Text Mining Where You Eat**

**Introduction**

Yelp’s crowd sourced method of business evaluations has grown exponentially sense its inception. Notwithstanding, Yelp’s recommendation tools are lacking. It uses other reviews to identify a business’s popularity and also promotes the most ‘useful’ or ‘helpful’ reviews of a given business. To accurately predict your personal feelings towards a business Yelp relies on you being a member of the majority who hypothetically will also have a positive opinion of a popular business. Nowhere does Yelp try to personally predict what businesses you may like based on your own unique preferences.

Using Yelp’s publicly available academic dataset and focusing specifically on restaurants I recognized this issue. I wondered if Yelp could reference a person’s favorite restaurants from one city and find restaurants based on similar criteria and reviews in a new city.

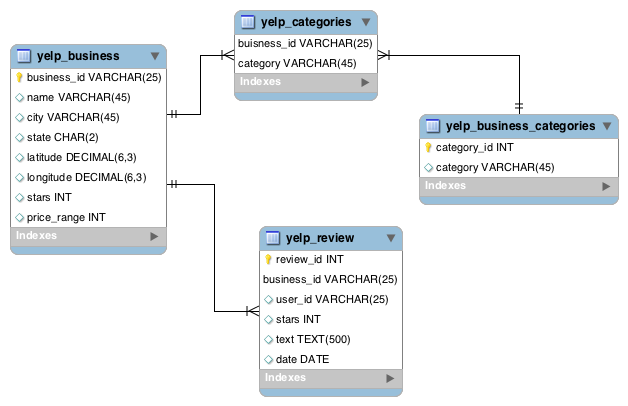
Since Yelp hosts an annual challenge with their dataset, there is plenty of work done similar to this project. And even if there were not a publicly available dataset there would still be a ton of interest in Yelp. Businesses across the nation care about there Yelp rating and millions of consumers base where buy, shop and eat on the quality of a businesses Yelp ratings.

Some examples of these projects include, Chee Hoon Ha, a Stanford student used the Yelp dataset to try and [predict](http://cs229.stanford.edu/proj2014/Chee%20Hoon%20Ha,%20Yelp%20Recommendation%20System%20Using%20Advanced%20Collaborative%20Filtering.pdf) a given users rating for a certain restaurant based on similar users. In another [project](http://courses.cs.washington.edu/courses/cse544/13sp/final-projects/p09-tingtinz.pdf) Tingting Zhang and Yi Pan and the University of Washington used the Yelp data to analyze the success (star rating) of restaurants based on other data such as location and yelp check-ins. Chen Li and Jin Zhang at Stanford also used the Yelp dataset to mine text sentiment of reviews in order to [predict](http://cs229.stanford.edu/proj2014/Chen%20Li,%20Jin%20Zhang,%20Prediction%20of%20Yelp%20Review%20Star%20Rating%20using%20Sentiment%20Analysis.pdf) the star rating. It’s obvious that there is interest in this topic because of the wide use of Yelp and business potential of any useful tool.

Personalized recommendation tools have been implemented across the web (Amazon, iTunes, Facebook friends just to name a few) and Yelp has the capability to continue this trend with there data set using the category of restaurants, price range, location, and star rating of each restaurant. Categories refer to the type of the restaurants such as Mexican, American, Italian, etc.. These can help recommend restaurants because they indicate a person’s specific taste. If their three favorite restaurants in their hometown are Italian restaurants then there is good reason to recommend an Italian restaurant. Price range also can indicate a persons preference and whether or not the price of food is strongly important to them. And star rating will serve as a good baseline because no one wants to eat at a restaurant with overwhelmingly bad reviews. Finally, location will be used to gather data from cities specifically.

**Data collection, cleaning, and storage**

To start I downloaded the Yelp academic dataset provided by Yelp which can be found [here](https://www.yelp.com/dataset_challenge). There was no reason to explore other data sources since Yelp was providing the data directly. Yelp gives several JSON files in a zip file including data on there social network, but I was only interested in the “review” and “business” files for this project. First, I created a Entity Relationship Diagram to map out where my data would go in the new database. From the JSON business file we will want the business\_id’s, the name of the business, the city, state, latitude, longitude, average star (rounded) and price range. Also in this database we will want to grab the categories, but in order to stick to 3rd Normal Form we will place that data in another table. We will have a yelp\_business\_category table, which will hold all possible categories and will give them an artificial primary key called category\_id. To connect these two tables we will have a table called yelp\_categories, which will hold each business’s unique categories. Both values can be repeated, but not in the same row. One row will be repeated since it is a composite key. From the review JSON file we will want to grab the review\_id, business\_id, user\_id, stars, text, and date. The review\_id is an artificial primary key since there was not a natural key. Below the ERD there is a thorough data dictionary explaining each column of the database.



|  |  |
| --- | --- |
| yelp\_business |  |
| business\_id | Encryptyed id for each business in yelp’s database. |
| name | Name of the place of business |
| city | City where the business is located. |
| state | State where the business is located |
| latitude | Latitude of the business. |
| longitude | Longitude of the business. |
| stars | Average star rating of the business |
| price\_range | Price range scale used by yelp (1-4) |

|  |  |
| --- | --- |
| yelp\_categories |  |
| business\_id | Encryptyed id for each business in yelp’s database. |
| category | Categories that pertain to the business and the type of business, food served, and atmosphere. |

|  |  |
| --- | --- |
| yelp\_business\_categories |  |
| category\_id | Artificial primary key |
| category | Categories that pertain to the business and the type of business, food served, and atmosphere. |

|  |  |
| --- | --- |
| yelp\_review |  |
| review\_id | Artificial primary key used to give each row a unique identifier |
| business\_id | Encryptyed id for each business in yelp’s database. |
| user\_id | Encryptyed id for each user in yelp’s database. |
| Stars | Star rating given to business by user |
| Text | Text that was written in the review |
| date | Date of review |

To import these into the database I ran several different pieces of code, which can be found below.

Each python script imported data into a different database (except the last one which was an update on the reviews database.) They each follow the same basic structure. Import the necessary libraries, connect to the database, set up an insert statement, find the necessary data from the json file, then execute the json and import the data into the database.

yelp\_business

1. # -\*- coding: utf-8 -\*-
2. """
3. Created on Wed Nov 30 18:14:42 2016
5. @author: Wesley
6. """
8. **import** json
9. **import** pymysql
10. **import** sys
12. #grab password from commandline
13. password = sys.argv[1]
15. # Open database connection
16. db = pymysql.connect(host='cs.elon.edu',
17. db='wupham',
18. user='wupham',
19. passwd=password,
20. port=3306,
21. charset='utf8mb4',
22. autocommit=True)
24. cursorSelect = db.cursor()
25. insertQuery = "INSERT INTO yelp\_business (business\_id, type, name, city, state, latitude, longitude, stars) VALUES (%s, %s, %s, %s, %s, %s, %s, %s)"
27. with open('yelp\_business.json') as myfile:
28. **for** line **in** myfile:
29. data = json.loads(line)
30. busType = data.get("type")
31. businessID = data.get("business\_id")
32. name = data.get("name")
33. city = data.get("city")
34. state = data.get("state")
35. latitude = data.get("latitude")
36. longitude = data.get("longitude")
37. stars = data.get("stars")
38. #print(busType, businessID, name, city, state, latitude, longitude, stars)
39. cursorSelect.execute(insertQuery,(businessID, busType, name, city, state, latitude, longitude, stars))

yelp\_categories

1. # -\*- coding: utf-8 -\*-
2. """
3. Created on Wed Nov 30 18:34:09 2016
5. @author: Wesley
6. """
8. # -\*- coding: utf-8 -\*-
9. """
10. Created on Wed Nov 30 18:14:42 2016
12. @author: Wesley
13. """
15. **import** json
16. **import** pymysql
17. **import** sys
19. #grab password from commandline
20. password = sys.argv[1]
22. # Open database connection
23. db = pymysql.connect(host='cs.elon.edu',
24. db='wupham',
25. user='wupham',
26. passwd=password,
27. port=3306,
28. charset='utf8mb4',
29. autocommit=True)
31. cursorSelect = db.cursor()
32. insertQuery = "INSERT INTO yelp\_categories (business\_id, category) VALUES (%s, %s)"
34. with open('yelp\_business.json') as myfile:
35. **for** line **in** myfile:
36. data = json.loads(line)
37. businessID = data.get("business\_id")
38. categories = data.get("categories")
39. **if** 'Restaurants' **in** categories:
40. **for** category **in** categories:
41. #print(businessID, category)
42. cursorSelect.execute(insertQuery,(businessID, category))

yelp\_business\_categories

1. # -\*- coding: utf-8 -\*-
2. """
3. Created on Wed Nov 30 18:55:35 2016
5. @author: Wesley
6. """
8. **import** json
9. **import** pymysql
10. **import** sys
12. #grab password from commandline
13. password = sys.argv[1]
15. # Open database connection
16. db = pymysql.connect(host='cs.elon.edu',
17. db='wupham',
18. user='wupham',
19. passwd=password,
20. port=3306,
21. charset='utf8mb4',
22. autocommit=True)
24. cursorSelect = db.cursor()
25. insertQuery = "INSERT INTO yelp\_business\_categories (category\_id, category) VALUES (%s, %s)"
27. categoryList = []
29. with open('yelp\_business.json') as myfile:
30. **for** line **in** myfile:
31. data = json.loads(line)
32. categories = data.get("categories")
33. **if** 'Restaurants' **in** categories:
34. **for** category **in** categories:
35. **if** category **not** **in** categoryList:
36. categoryList.append(category)
38. counter = 0
39. **for** item **in** categoryList:
40. counter = counter + 1
41. cursorSelect.execute(insertQuery,(counter, item))

yelp\_reviews

1. # -\*- coding: utf-8 -\*-
2. """
3. Spyder Editor
5. This is a temporary script file.
6. """
7. **import** json
8. **import** pymysql
9. **import** sys
11. #grab password from commandline
12. password = sys.argv[1]
14. # Open database connection
15. db = pymysql.connect(host='cs.elon.edu',
16. db='wupham',
17. user='wupham',
18. passwd=password,
19. port=3306,
20. charset='utf8mb4',
21. autocommit=True)
23. cursorSelect = db.cursor()
24. insertQuery = "INSERT INTO yelp\_review (review\_id, type, business\_id, user\_id, stars, text, date) VALUES (%s, %s, %s, %s, %s, %s, %s)"
26. yelpList = []
27. counter = 0
28. with open('yelp\_academic\_dataset\_review.json') as myfile:
29. **for** line **in** myfile:
30. data = json.loads(line)
31. yelpType = data.get("type")
32. businessID = data.get("business\_id")
33. userID = data.get("user\_id")
34. stars = data.get("stars")
35. text = data.get("text")
36. date = data.get("date")
37. cleanText = text.replace('\t',' ').replace('\n',' ').replace('\r',' ')
38. counter = counter + 1
39. cursorSelect.execute(insertQuery,(counter, yelpType, businessID, userID, stars, cleanText, date))

adding price column to the yelp reviews

1. # -\*- coding: utf-8 -\*-
2. """
3. Created on Thu Dec  1 12:54:34 2016
5. @author: Wesley
6. """
8. **import** json
9. **import** pymysql
10. **import** sys
12. #grab password from commandline
13. password = sys.argv[1]
15. # Open database connection
16. db = pymysql.connect(host='cs.elon.edu',
17. db='wupham',
18. user='wupham',
19. passwd=password,
20. port=3306,
21. charset='utf8mb4',
22. autocommit=True)
24. cursorSelect = db.cursor()
25. updateQuery = "UPDATE yelp\_business SET price\_range = %s WHERE business\_id = %s"
27. with open('yelp\_business.json') as myfile:
28. **for** line **in** myfile:
29. data = json.loads(line)
30. price = data.get("attributes")
31. businessID = data.get("business\_id")
32. priceRange = price.get("Price Range")
33. #print(priceRange, businessID)
34. cursorSelect.execute(updateQuery,(priceRange, businessID))

You can view these files in more depth in the zip file, which this word document was included in. I also have a folder called testFiles, which contains 50 lines of each of the JSON files in order to test the python scripts more effectively.