Retrieving, Ranking, Collecting Restaurants based on Yelp Reviews

Team Members

1. Akshay Punhani
2. Rajvardhan Oak

Project Motivation & Goal

As regular Yelp users (a.k.a foodies) and compulsively obsessed with reading reviews before trying out new restaurants/bars, we believe that categorizing them into themes/collections would allow for easier selection. Yelp already has a layer of social discovery, but we want to go a step further and provide information that foodies want, but may not necessarily want to search for. For example, a user gets to see the collection of restaurants under “restaurants with a view” if he/she wants to go eat and drink at someplace scenic.

Our aim is to build an information organization and retrieval system for foodies to search for restaurants on Yelp based on **similarity of user reviews**. Although Yelp provides some minimal categorization, we plan to add some additional features that are inferred from the reviews. These categories exist implicitly, but we make them explicit: such as ‘Date Places’, ‘Sports Bars’, ‘Good Discounts’.

Yelp Dataset

The dataset we leveraged in our analysis is a subset of Yelp's businesses and reviews data. The original dataset approximately contained 5,200,000 user reviews, information on 174,000 businesses in 10 metropolitan areas. The dataset could be found on https://www.yelp.com/dataset. A sample of the business and review datasets is attached for reference in Appendix A.

Resources Organized:

Restaurants/bars and their **user reviews** on Yelp are the principal resources we organized. We used semantic analysis on user reviews to classify and organize resources.

The Business data set contains information about the business, its associated category, number of reviews, etc. The reviews data set contain user reviews and ratings. Thus, we have the following main resources:

1. *Reviews:* These are the comments, reviews, testimonials submitted by Yelp users for each restaurant. A restaurant may have more than one review.
2. *Ratings:* Along with the review, users rate the restaurant out of 5 (1 being the worst, 5 being the maximum).
3. *Categories:* Every entry in the business dataset has a list of categories associated with it. This identifies the type of business or other specific characteristics.

Resource Organization & Structure

1. ***Data Cleaning:*** This consisted of the following steps:
    A. Dimensionality Reduction: We reduced the number of dimensions by eliminating the columns (such as neighborhood etc.) which were not required from both the data sets.

    B. Missing Values: Some values were Null or NaN. We dropped such entries which had missing values.

    C. Data Reduction: We included only those businesses which were tagged as 'Restaurants'. In addition, the restaurants with few reviews can give us less information than the ones with many reviews. On these lines, we clipped our data and included only those restaurants which had at least a certain number of reviews. A graph was plotted to make an informed decision pertaining to how many minimum reviews each restaurant should have in our final analysis. As can be observed in the graph, the cut off point for the number of reviews where there are maximum restaurants is ~70.

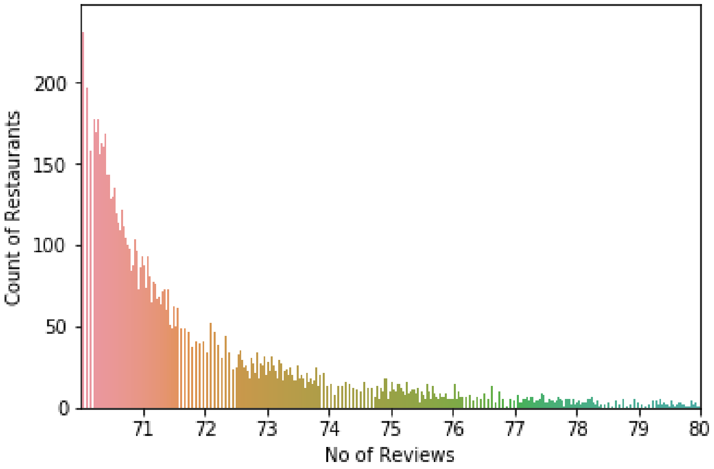


Figure [1] - Distribution of Restaurants based on number of user reviews

2. ***Data Structure:*** We **merged (inner join)** the two data sets on the common column, that is, ‘Business_ID’, which was a **primary key** in the Business data set. Since each business has many reviews, the linkage between the resources or the data frames could be defined as **one to many relationship**. Each row in the final dataframe points to a unique review. In the end, our dataset reduced from about 5.2 million to just 2.3 million reviews. The final cleaned, merged dataset consisted of the following columns (attributes):

    1. *Business_id* - The unique business id for the restaurant/bar
    2. *Name* - Name of the establishment/restaurant
    3. *Address* - Complete address of the restaurant
    4. *City* - City where the restaurant is located
    5. *State* - State where the restaurant is located
    6. *Postal_code* - Postal code of the restaurant
    7. *Stars* - Average rating of the restaurants
    8. *Review_Rating* - Rating given for that particular review
    9. *Categories* - Categories under which the restaurant falls into - Pizza, event planning, italian etc.
    10. *Review_id* - A unique review id for each review
    11. *User_id* - A unique id of the user who posted the review
    12. *Text* - The text of the review
    13. *Date* - Date when the review was posted on Yelp

Organizing System

Our organizing system leveraged the reviews and the review ratings provided by the user on Yelp. First, we created a search engine which took a list of words to return the reviews that contain *any* of those words. The goal was to create customized collections of restaurants based on specific keywords which we decided based by manually looking through the reviews and basis the frequency of words, which were common to that particular theme or collection. For example, for romantic places and date spots, we manually labelled the reviews by searching for specific keywords such as ('romantic, couple, date, tinder’) since these were the most frequently occuring words in the user reviews of these restaurants. At this time, we created four new categories in line with our aims of bringing out ‘hidden’ or implicit categories.

A. Sports Bar: This category defines whether the restaurant is a good sports bar or not.

B. Has_Waffles: This specifies whether the restaurant serves waffles or not.

C. Review Score: This specifies that score that has been assigned to each restaurant basis the average rating of the restaurant and the number of reviews it has received till date.
D. Date Spot: This category defines whether the restaurant is a good date/romantic spot or not.

Retrieval System

As part of the retrieval system, we gave the users the option of choosing a restaurant basis the combination of keyword and query, and access to collection/themes of restaurants based on our analysis.

1. **Based on query:** Users can search based on more complex queries such as *'Good clam chowder near golden gate bridge'*
2. **Collections:** Users can view standard collections/categories - leveraging user reviews, ratings and some specific tips such as "romantic places"

Retrieving Based on Review/Query Similarity

As a problem statement, we wanted to return restaurants that had similar reviews as compared to the query typed in by the user. For example "nice pizza place in Pittsburgh near downtown", should return the pizza places in Pittsburgh near downtown. For the query based searching, we used Tf-idf and the cosine similarity.

TF-IDF reflects how important a word is to a document in a collection or corpus.It adjusts the raw term frequency of a word by taking into account how frequently each word occurs in general (the Document Frequency). Inverse Document Frequency is usually the log of the number of documents divided by the number of documents (total reviews) the word occurs in. TF-IDF is often used as a weighting factor in searches of information retrieval, text mining, and user modeling. The tf–idf value increases proportionally to the number of times a word appears in the document and is offset by the number of documents in the corpus that contain the word, which helps to adjust for the fact that some words appear more frequently in general.

Cosine Similarity measures the cosine of the angle between two vectors. In our case, we use the vector space model where each query/set of words can be presented as a vector whose direction is determined on a set of the TF-IDF values in the space. Because vectors have direction as well as magnitude, the cosine similarity is a better distance metric than euclidean distance (which simply calculates the distance between two points in space).

In our system, the following happens when a query is entered:

1. The query is converted to a vector using TF-IDF measure for each word.
2. Cosine similarities are computed between the vector created in Step 1 above and the documents in our dataset (reviews).
3. Return top 5 reviews (i.e. those with the maximum cosine similarity to the query) along with the restaurant names.

Retrieving Based on Collections

Yelp currently provides only a limited set of collections. These are typically centered around cuisine such as Mexican, Italian, Chinese, etc. We decided, however, to create custom collections which would refer to hidden but meaningful characteristics such as 'Date Spots', 'Sports Bars', etc. as described in the organizing system above.

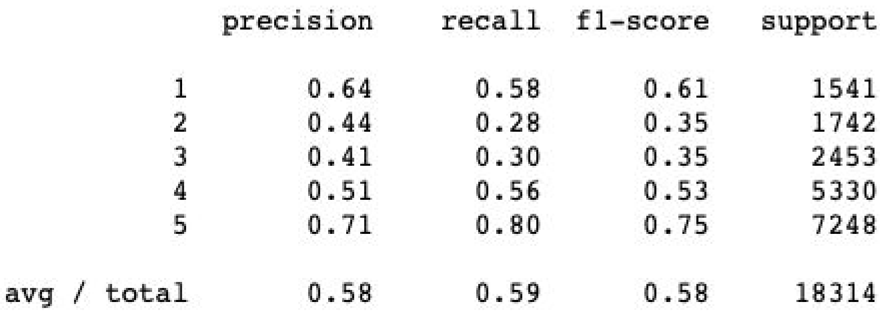
To classify restaurants under these custom collections, we parsed the reviews and looked for certain keywords. For example, for a restaurant to belong to the 'Date Spot' collection, we used the keyword set as {love, romantic, couple, tinder, romance, anniversary, proposal}. These keyword sets were built by intuition.

After annotating 30% of the dataset using the method described, we applied machine learning algorithms to predict the collection membership for the remaining data. We achieved high accuracies of 87% (decision tree) and 91%(random forest).

Sentiment Analysis

As an additional functionality, we also implemented a review-based sentiment analysis system. **A possible use case for this would be to predict the expected rating based on the review**. This would be a good method to normalize the ratings and ensure the review-rating consistency.

We then used **Naive-Bayes classifier** to predict the rating. We achieved the below precision, recalls for each of the 5 ratings.

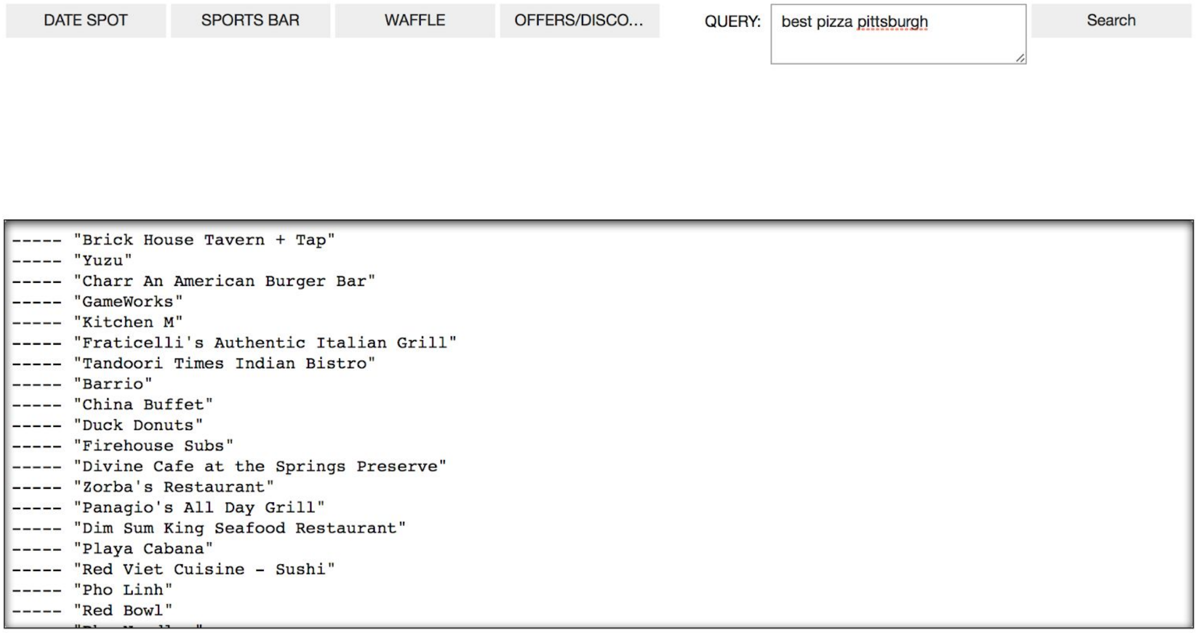


```
             precision    recall  f1-score   support

          1       0.64      0.58      0.61      1541
          2       0.44      0.28      0.35      1742
          3       0.41      0.30      0.35      2453
          4       0.51      0.56      0.53      5330
          5       0.71      0.80      0.75      7248

avg / total       0.58      0.59      0.58     18314
```

Figure[2]- Confusion Matrix

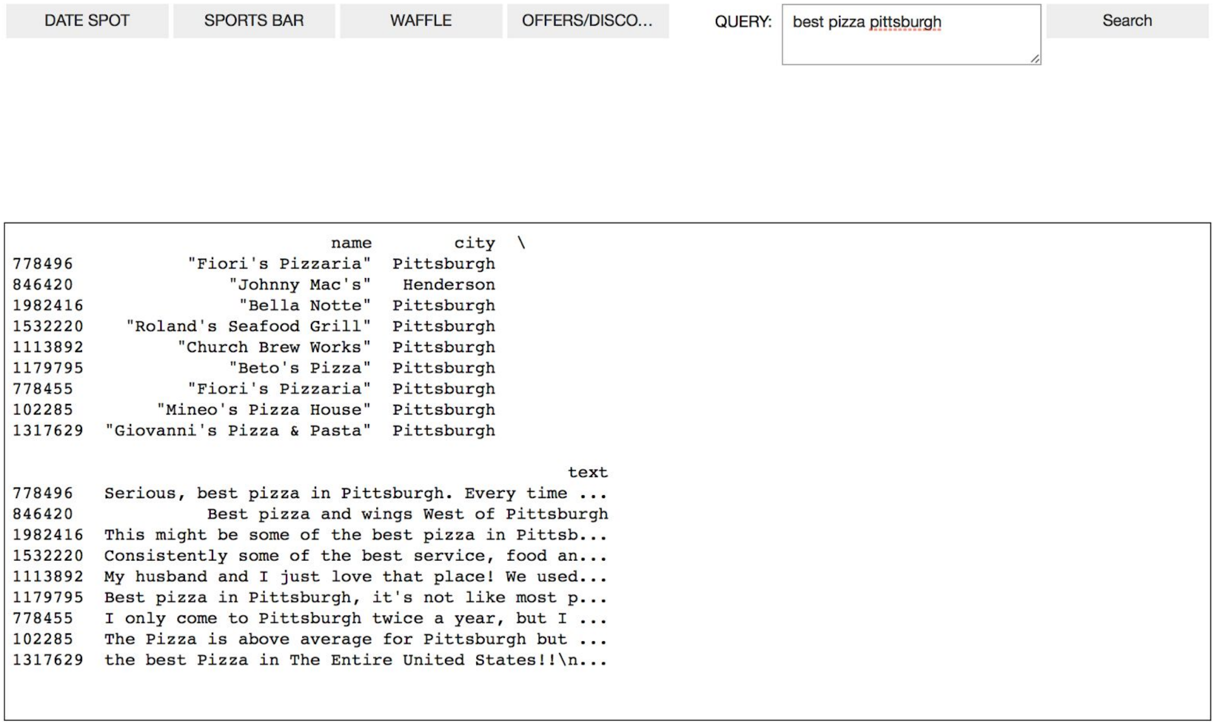
Implementation

Figure [3] represents the result when the user tries to query for a particular custom collection. Figure [4] represents the result of a natural language query.



```
----- "Brick House Tavern + Tap"
----- "Yuzu"
----- "Charr An American Burger Bar"
----- "GameWorks"
----- "Kitchen M"
----- "Fraticelli's Authentic Italian Grill"
----- "Tandoori Times Indian Bistro"
----- "Barrio"
----- "China Buffet"
----- "Duck Donuts"
----- "Firehouse Subs"
----- "Divine Cafe at the Springs Preserve"
----- "Zorba's Restaurant"
----- "Panagio's All Day Grill"
----- "Dim Sum King Seafood Restaurant"
----- "Playa Cabana"
----- "Red Viet Cuisine - Sushi"
----- "Pho Linh"
----- "Red Bowl"
```

Figure [3] - Custom Collection *Date Spot*



best pizza pittsburgh

```
                              name        city  \
778496          "Fiori's Pizzaria"  Pittsburgh
846420              "Johnny Mac's"   Henderson
1982416              "Bella Notte"  Pittsburgh
1532220   "Roland's Seafood Grill"  Pittsburgh
1113892        "Church Brew Works"  Pittsburgh
1179795             "Beto's Pizza"  Pittsburgh
778455          "Fiori's Pizzaria"  Pittsburgh
102285         "Mineo's Pizza House"  Pittsburgh
1317629  "Giovanni's Pizza & Pasta"  Pittsburgh

                                                    text
778496   Serious, best pizza in Pittsburgh. Every time ...
846420             Best pizza and wings West of Pittsburgh
1982416  This might be some of the best pizza in Pittsb...
1532220  Consistently some of the best service, food an...
1113892  My husband and I just love that place! We used...
1179795  Best pizza in Pittsburgh, it's not like most p...
778455   I only come to Pittsburgh twice a year, but I ...
102285   The Pizza is above average for Pittsburgh but ...
1317629  the best Pizza in The Entire United States!!\n...
```

Figure [4] - Result for Query

Conclusion

We experimented and tried to derive meaningful knowledge from the Yelp Dataset. Using keyword based search, we created custom collections for inherent features. We further used TF-IDF and Cosine similarity to return the most relevant restaurants based on a user's query. In addition, we used Naive-Bayes classifier to predict the rating based on the review text.

Acknowledgements

We would like to thank Prof. David Bamman and Andrew Chong for their advice and insightful comments at various stages of this project.

Appendix-A

Columns

#	Column Name	Sample Field
1.	business_id	**Apn5Q_b6Nz61Tq4XzPdf9A**

2.	name	**Minhas Micro Brewery**
3.	neighborhood	**NAN**
4.	address	**1314 44 Avenue NE**
5.	city	**Calgary**
6.	state	**AB**
7.	postal_code	**T2E 6L6**
8.	latitude	**51.0918130155**
9.	longitude	**-114.031674872**
10.	stars	**4**
11.	review_count	**24**
12.	is_open	**1**
13.	categories	**Tours, Breweries, Pizza, Restaurants, Food, Hotels & Travel**
14.	hours	Monday:**8:30-17:0** Tuesday:**11:0-21:0** Wednesday:**11:0-21:0** Thursday:**11:0-21:0** Friday:**11:0-21:0** Saturday:**11:0-21:0**