INDIANA UNIVERSITY BLOOMINGTON

FINAL REPORT

Yelp Dataset Challenge

Authors:
Aishwarya Pramod Dhage
Nawaz Hussain Khazielakha
Praneta Vishwanath Paithankar

Faculty:
Xiaozhong Liu

A final report submitted in the fulfillment of the requirement of course ILS - Z534 Information Retrieval

in

School of Informatics, Computing and Engineering

December 12, 2017

INDIANA UNIVERSITY BLOOMINGTON

Abstract

Yelp Dataset Challenge

by
Aishwarya Pramod Dhage
Nawaz Hussain Khazielakha
Praneta Vishwanath Paithankar

Given was the businesses and reviews data from Yelp. It contained well defined attributes which qualified each business(on rating as a metric) and also contained reviews(textual and rating points) from the users who rated the businesses. Supporting data of pictures, tips and check-in timings were also provided. Our tasks(were to analyze the given data (for business: restaurants) and present our findings which could help Yelp and the related businesses in improving their quality.

Task 1: Recommend businesses to users. We chose two approaches for this, information retrieval and item based collaborative filtering approach. Data was divided into two sets training and test data. Post processing the data, we have used standard analyzer and BM25Similarity to retrieve top recommendations to the user. To test the accuracy of the retrieved results, we have used Mean Absolute Percentage Error(MAPE), Precision and Recall evaluation metrics's results and tested it against our training data.

Task 2: Recommend locations to an user who wants to open a new restaurant of a particular cuisine. User should be able to understand the existing demand of different cuisines in a city, visually. We have statistically analyzed the ratings, assigned weights to each review, and plotted user friendly heat-maps depicting the popularity distribution of a cuisine across the city. Using this we also extracted popularity of restaurants. To test the accuracy of the popularity of the suggested places of popular cuisines, we searched for restaurants which actually served those cuisines with high rating(rating is given by the user at the time of initial plotting and testing)

Contents

Chapter 1

Introduction

1.1 Yelp.com

Yelp is a popular(among users) online portal which maintains information about businesses, reviews, rating and user profiles who rate the businesses. A yelp-user (yelper) can extract the list of businesses of a particular rating and also provide "rich" feedback about any business. To increase the popularity of each business, the business owners tend to provide multiple details of their business,such as address, longitude, latitude, category, amenities and of course their name.

Each user upon registration also provide their details and add friends as they spend much time on the portal. Users also review the business they have visited and can also add photos to it. All of this review data is stored at yelp databases and each business is evaluated (rated in a point scale of 5). Moreover, each review provided by the user is also rated by other users if they find it useful.

1.2 The data set challenge

What if the user (new or not) wants to find out about new popular businesses in town? Will he be able to extract this information from Yelp website? Even if the user is provided with a list of restaurants, will he be able to gauge the its quality quantitatively?

Or what if the user (new/existing) wants to setup a new business(restaurant) in a city? Will he be able to locate the popular areas visited by users(reviewers) for a particular cuisines without visiting the places in person? This is more or less the the major functionality of Yelp.com.

In this data-set challenge, we are given access to business, review, checkin, and photos data of specific cities around the world. Considering ‘restaurants’ as the limiting factor of the data set, we developed a recommendation system addressing the challenges mentioned in the start of this section.

The data was shared in json files form Yelp. We stored the data in MongoDB to the store data locally.

Chapter 2

Task 1

In this task, we are recommending the businesses to users. We are using two approaches to recommend businesses.

- Information Retrieval
- Collaborative Filtering

2.1 Data Refinements and Constraints

Input Data was restricted to city 'Charlotte' and populated in mongoDB. MongoDB was chosen over MySQL implementation of the dataset because MongoDB is a schemaless DB and this enables us to store and retrieve data without any data type hassle. Moreover, to retrieve the data from MongoDB, we need smaller queries when compares to MySQL queries.

For the task1,

- We are considering businesses whose total reviews are equal to and greater than 100.
- Users who have given at least 20 reviews for the businesses that we have refactored.
- We divided above data into two data i.e. training data and testing data.
- To divide data,we considered the review date.If review has written before 1-1-2015,we are considering it as training data.If review has written on or after 1-1-2015, we are considering it as testing data.

2.2 Method: Information Retrieval

2.2.1 Proposed Algorithm

- We are considering user's review to retrieve the information
- In order to find the relevance of user review with reviews of restaurants,we are dividing reviews in following way:
  - Bags of words

  - Noun
  - Noun + adjectives
  - Adjectives

- We are using all reviews of users present in training data to understand the preferences of users.

- In this approach,we used Lucene to retrieve the information

2.2.2 Experiment Design

- In order to generate index, we are creating document for each restaurant.

- We are storing all reviews for that restaurant in the document. Here we are using POS tagging to find noun,adjectives etc.If we are considering only nouns then we are only storing noun in reviews.

- We used opennlp library to find the POS tagging of words.

- After generating index using Lucene, we are searching the index using query. To generate query, we are considering all reviews of user.Here we are also using opennlp library for pos tagging.In case of noun only ,we are only adding noun in query.

- To get the top recommendation for each user ,we are using standard analyzer and BM25Similarity.

2.3 Method: Collaborative Filtering

To suggest businesses to users based on their rating we are using Collaborative Filtering. There are namely two types of Collaborative Filtering-User-based and item-based , in our project we are using item based collaborative filtering. Since user based collaborative filtering is computationally expensive and also user profile changes quickly we cannot use user based CF efficiently for a data-set like Yelp.

2.3.1 Experiment Design

- In this approach we are using Mahout to implement item-based collaborative filtering.

- It takes a .csv input with three columns which are user ids, business ids and user ratings given to businesses.

- We , then used this input file to generate a model.

- To calculate item to item(business to business) similarity we are using Pearson Correlation Similarity.

- Using recommender builder we have generated all the recommendations for every user id based on business similarity.

2.4 Evaluation Metrics

2.4.1 Mean Absolute Percentage Error

$M = \frac{100}{n}\Sigma_{t=1}^{n}\frac{|A_t-F_t|}{|A_t|}$

2.4.2 Precision

$Precision = \frac{\{relevant \quad documents\}\cap\{retrieved \quad documents\}}{\{retrieved \quad documents\}}$

2.4.3 Recall

$Recall = \frac{\{relevant \quad documents\}\cap\{retrieved \quad documents\}}{\{relevant \quad documents\}}$

2.5 Evaluation Table

Consolidated results of the experiment.

	CF	IR (noun + adj)	IR noun	IR adj	IR bag of words
Recall - Top 25	0.08	0.09	0.08	0.07	0.09
Precision - Top 25	0.04	0.05	0.04	0.05	0.04
Recall - Top 50	0.15	0.17	0.17	0.15	0.19
Precision - Top 50	0.04	0.04	0.05	0.05	0.04
Recall - Top 100	0.31	0.30	0.28	0.30	0.34
Precision - Top 100	0.04	0.04	0.04	0.05	0.04

Here IR is Information retrieval
Mean Absolute Percentage Error (MAPE):

Recommendations	CR	IR(noun + adj)	IR noun	IR adj	IR bag of words
25	92.45	91.61	90.72	92.12	91.75
50	85.06	84.41	82.39	86	80.07
100	69.45	72.65	74.49	68.07	74

Here CR is collaborative filtering

2.6 Conclusion And Future Work

In this project we found that Information retrieval approach is giving us better results than Collaborative filtering. This is because Information retrieval is considering reviews for giving recommendation while collaborative filtering is using just rating. Since rating can be misleading at times , using reviews for providing recommendation is better option.

For future work we feel that, extracting sentiments of users through reviews and tips and giving weight-age to each review accordingly will improve accuracy. In case of collaborative filtering we can normalize rating using this weight-age calculation.

Chapter 3

Task 2

Recommend locations to an user who wants to open a new restaurant of a particular cuisine. User should be able to understand the existing demand of specific cuisines in a city, visually. Finally, it is a business decision of the user to setup a business in a place already saturated with a particular cuisine or set it up at a new place altogether (This is not in the scope of this recommendation system)

3.1 Data Refinements and Constraints

Input Data was restricted to city 'Charlotte' and populated in mongoDB. MongoDB was chosen over MySQL implementation of the dataset because MongoDB is a schema-less DB and this enables us to store and retrieve data without any data handling hassle. Moreover, to retrieve the data from MongoDB, we need smaller queries when compares to MySQL queries.
Moreover, the DB is lightweight, performs faster and also has simpler view restrictions which enables it for a rapid development.

For the Task 2,

- We are considering all the businesses irrespective of their review count. This is to include newly opened restaurants which are popular but have not many reviews logged on them.
- Users review is weighted on the votes each review gets. Each review in the dataset is associated with multiple categories. We have considered only 'useful' tag as votes. We have neglected others as other votes on other categories does not add any qualitative value to the review of the user.
- We have not divided the data for this task, models from reviews of real-time users. The accuracy of output-ed topics(or cuisines) was matched with the 'category' attribute associated to each restaurant.

3.2 Method: Statistical Analysis

3.2.1 Proposed Approach

- Given the reviews of each restaurant belonging to city 'Charlotte', we extracted the topics, assigned them a weight by statistical analysis and plotted their heat map.

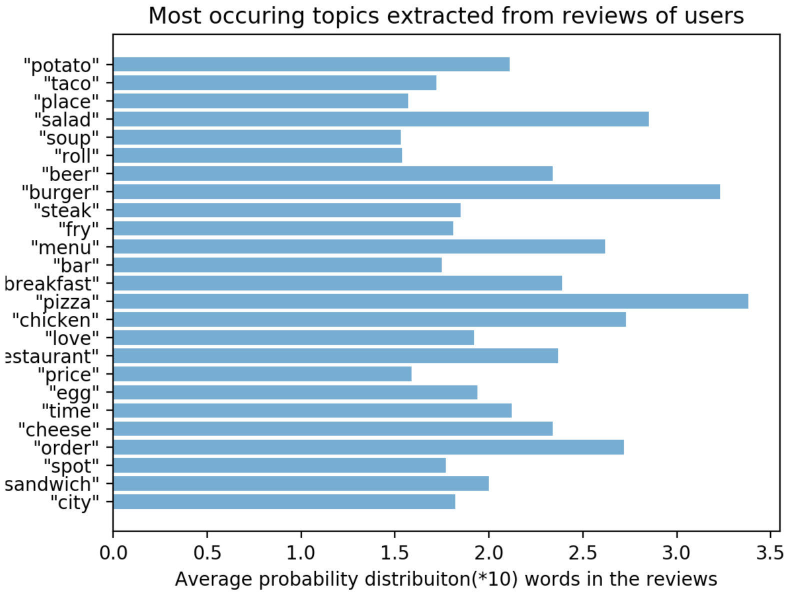


FIGURE 3.1: Probability distribution(*10) of top 25 topics extracted from reviews of the users whose value is more than 1.5

- For evaluation, we calculated the ‘hit-ratio’ for each topic extracted from the reviews, and check it against the category attribute present in each business(restaurant). If the category exists then the proposed topic by the recommendation system is valid.

3.2.2 Experiment

- All the businesses belonging to the city = “Charlotte” were uploaded in the database. All the reviews addressing the businesses in the city Charlotte were uploaded in the DB. (1LoadCorpus.py)

- A total of 2327 businesses were present in the city Charlotte. 141281 reviews were recorded by the users for 2327 businesses.

- Reviews were tokenised, and nouns and plural nouns were extracted as bag of words for each review and stored in another separate collection.

- Natural Language Toolkit was used for the lexical analysis and parts of speech tagging of the words in the reviews(Refer 3.1).(2findTop60Topics.py)

- Corpora package of Genism Library is used to retain frequently occurring words from the review. Only top 10,000 recurring words from a total of 52,000 words were retained as considering more words would be eliminated in later stages of processing(at the time of weight calculation, the words with less weight are not selected naturally). These 10,000 words are considered as the bag of words. (4businessrRatings.py)

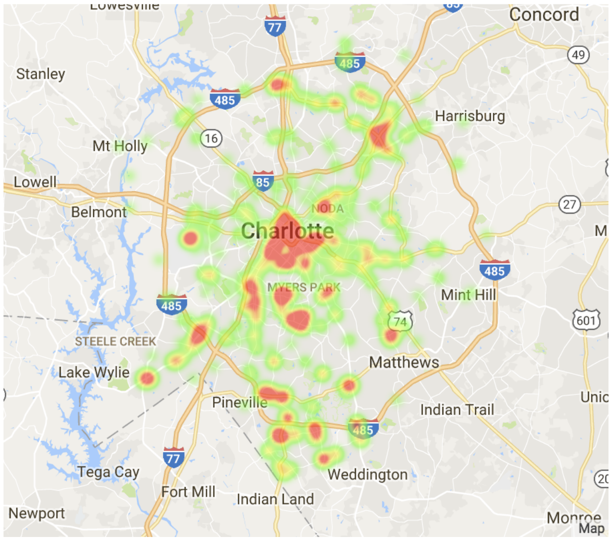


FIGURE 3.2: Heat Map of Places serving Salad with a user rating of more than 2.Topic‘Salad’ is extracted from examining reviews of the users.

- Probability distribution of the words along with their word id is stored in a dictionary. This is done using Latent Dirichlet Allocation (gensim.models.LdaModel) (4businessrRatings.py)
- Top Sixty topics with their probability distribution are collected. These are the keywords which are most frequently found in the user reviews. These also the words used for analysis (as in they will be proposed to the user). (5PopulateClassificationAndBusiness.py)
- All the words are extracted from the dictionary and their weight is calculated using their frequency count and probability distribution.
- Logic used here to calculate the ‘popularity’ of a topic is the following:
  - The words present in the review with higher vote describing the business with higher rating implies that the word is of higher importance.
  - Hence, the rating of the business is added to the word for each occurrence and then normalized with its occurrence-count to make the weight of the word comparable with other words.(4businessrRatings.py)
- We maintain a collection of word-ids, their final weights and their count.
- HeatMap is plotted for a sample Cuizine (say Sandwich and Salad) having an expected rating of say 2. This heatmap represents the places in Charlotte which serve Cuisines (Sandwich and Salad) and have a rating more than 2.(Refer 3.2 and **??**)(7Gmaps_HeatMap.ipynb)
- This HeatMap can be used by the user to judge the places which serve most popular (sample) cuisines. The user can now chose a place strategically, if they want to setup their own business with the sample cuisine or not.

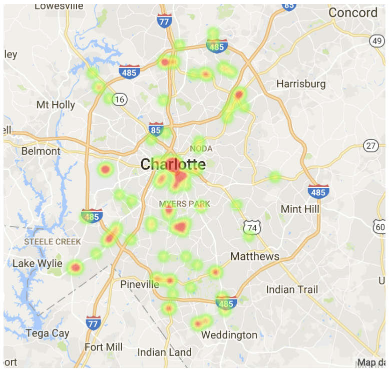
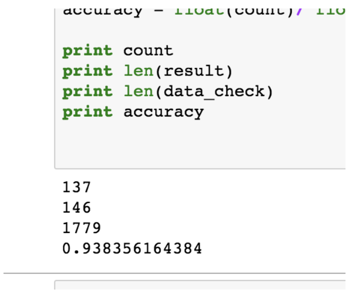


FIGURE 3.3: Heat Map of Places serving Salad with a business rating of more than 3



```
accuracy = float(count)/ flo

print count
print len(result)
print len(data_check)
print accuracy
```

```
137
146
1779
0.938356164384
```

FIGURE 3.4: Hit-Ratio of businesses serving salad retrieved from reviews vs category

3.3 Evaluation

- To Evaluate the accuracy of the HeatMap predicted by the program, we search for the restaurant which is actually serving (sample) category with an rating equal or higher than the one specified by the user initially.
- We select all those category-ied businesses and plot their heat map.
- We also Calculate a Hit-Ratio for every successful identification of the business retrieved from the review and one from the category (attribute of the dataset)(Refer **??** and **??** for Sandwich's accuracy, Refer 3.4 and 3.3 for Salad's accuracy)

3.4 Future Improvements

- Yelp can add another list of attributes in the Reviews collection where user can add the dish he is rating. This will help in better prediction of places based on their cuisine.
- Our program depends on the occurrence of category in the review. We can elaborate this approach to deduce category of the food being referenced in the review using stemming or other logical analyzers.
- We can also divide the data set by year and analyze the growth of restaurant across the city Charlotte. This can be very crucial in analyzing the city's food growth, We can then train models to predict how new businesses, if setup, may perform.