FINAL PROJECT REPORT - YELP DATASET CHALLENGE

ILS-Z 534 : SEARCH (by Xiaozhong Liu)

Arun Nekkalapudi, Chirag Galani, Janani Muppala, Keerthi Naredla

1. Introduction

The Yelp Dataset Challenge provided business, reviews , user data to research, analyse the data and figure out new patterns that may turn out be favourable to solve interesting problems. In this project we dealt with 2 interesting problems. First task is to recommend restaurants to user using 2 approaches and our second task is to predict most famous dishes from a restaurant. We used Mongodb, Lucene technologies and Java, Python and R programming languages.

Keywords : ***Content Based Filtering - Collaborative Filtering - Sentiment Analysis - Word Clouds***

2. Task 1: Recommending restaurants to user

Research Question: Our aim is to recommend business restaurants to each user using 2 methods and evaluate both the methods in order to determine which is a better approach to recommend business with data we have generated.

One approach is Content-based filtering, where User-profile is constructed from the user visited business and then business that match user-profile are recommended. Another approach is Collaborative-based filtering where top N nearest users to the target user are determined by using matrix factorization and then business visited by those nearest users expect those already visited by the target user are recommended.

2.1. Method 1 : Content-Based filtering

2.1.1. Index Generation

In yelp, business dataset provided have number of attributes, as it is difficult to handle such large amount of data we considered only specific attributes of Business Data such as Categories, price range, rating, location and indexed using Lucene.

Also the criteria is to index businesses which have reviews more than 20, to avoid the problem of invalid rating which can be caused if a business is highly rated by 2 or 3 duplicate users or owners themselves.

2.1.2. Query Formulation

First step is to generate a User profile of the target user (for whom we are recommending business). It consists of unique list of categories along with their average price range, list of user visited business that has this category and its frequency among the user visited business, based on which it is later sorted.

In order to generate this, the user id and user visited business list has to be determined as user visited business are not directly mentioned in the dataset provided by yelp. It is retrieved by joining review-id's in user dataset and business-id's in review data set.

After this, we got a list of categories, price-range, rating for each user visited business which is further used to generate user profile, as described above.

The query is formulated using Lucene Boolean Query as follows:
User profile AND stars (3.5 to 5) AND NOT business already visited by target user.

Sample Query:

```
+cat:Thai +prange:[2.0 TO 3.0] +stars:[5.0 TO 5.0] +stars:[4.5 TO 4.5]
+stars:[4.0 TO 4.0] -bid:ueoRWPGrSoZizl1ngBghqg -bid:2uRM8Et0uJVl8u1jSnmuKw
```

2.1.3. Business Recommendations

A business that is rated anywhere from 3.5 to 5, consisting of at least 1 category and if the price-range attribute of the business matches with the price -range specified in the query, then the business can be recommended.

For each category, we are recommending top 5 business that match all the criteria specified in the query.

2.1.4. Evaluation

The user visited business is divided into 70% as training and 30% as Testing data.

Evaluation metric chosen for this method is: Mean Average Precision.

MAP evaluation is based on number of business recommended converges with user visited businesses in testing data and also considering ranking of the matched business.

So we compute the precision at every correctly returned business restaurant, and then take an average precision at every match point: how many recommended business restaurants are relevant divided by the total business restaurants recommended up to that point.

MAP is just an extension, where the mean is taken across all AP scores for many queries.

$$\mathrm{MAP} = \frac{1}{|Q|} \sum_{q \in Q} \mathrm{AP}(q)$$

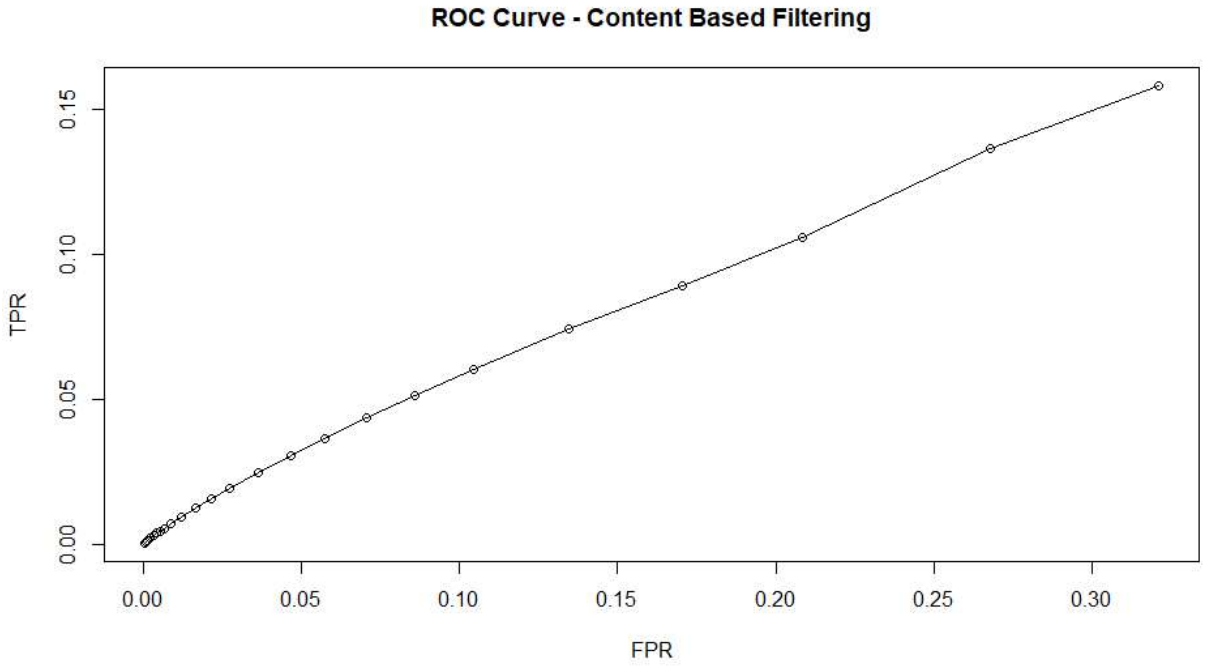
$$\mathrm{AP} = \frac{1}{|R|} \sum_{k=1}^{N} \mathrm{P}(k) \cdot \mathbb{1}_R(k)$$

We performed evaluation on 100 users , which implies 100 queries. The MAP for our recommendation model is …

With changes in training and testing data set and also by setting different similarities such as BM25, ... to recommend the business we got the following results.

User order	TP	FP	FN	TN	precision	recall	TPR	FPR
1	0.004359 82	3.784464 225	6.607573	6142.330 056	0.001151	0.000659 39	0.000659 39	0.000616
5	0.005922 96	5.365564 825	6.128258	6140.511 687	0.001103	0.000965 57	0.000965 57	0.000873
10	0.006690 12	6.106783 167	6.110585	6132.158 347	0.001094	0.001093 64	0.001093 64	0.000995
15	0.008586 18	8.063746 548	6.000476	6128.898 967	0.001064	0.001428 87	0.001428 87	0.001314
20	0.013946 56	14.23561 324	5.861679	6124.445 253	0.000979	0.002373 63	0.002373 63	0.002319
40	0.018676 37	20.43546 646	5.768372	6121.031 695	0.000913	0.003227 27	0.003227 27	0.003327
50	0.021946 67	26.68976 546	5.561103	6115.295 628	0.000822	0.003930 95	0.003930 95	0.004345
60	0.024686 34	32.64646 848	5.402323	6110.488 812	0.000756	0.004548 79	0.004548 79	0.005314
100	0.030648 48	41.19646 337	5.388169	6080.665 043	0.000743	0.005655 94	0.005655 94	0.006729
150	0.038070 56	52.81564 865	5.259503	6053.340 792	0.00072	0.007186 41	0.007186 41	0.00865
250	0.050646 54	72.09264 64	5.233654	5954.766 781	0.000702	0.009584 34	0.009584 34	0.011962
300	0.066467	98.73546	5.133734	5901.129	0.000673	0.012781	0.012781	0.016456

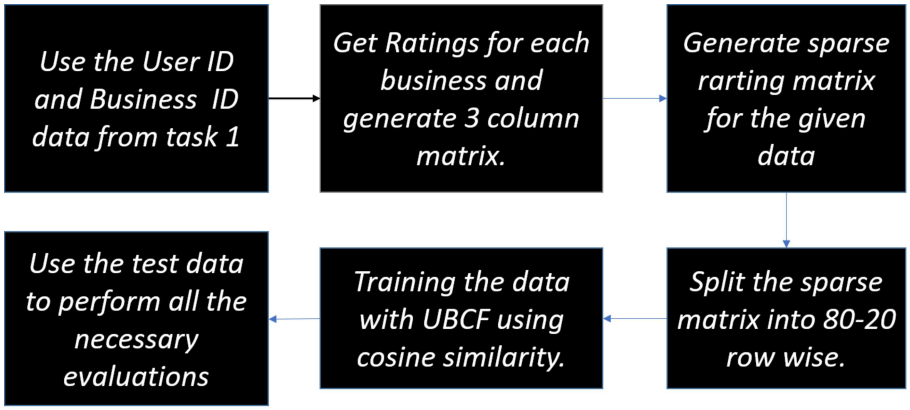
	43	59		088		7	7	
350	0.08176247	128.3575732	5.120148	5854.502299	0.000637	0.01571778	0.01571778	0.021454
400	0.10093287	160.9898224	5.111687	5763.868881	0.000627	0.01936317	0.01936317	0.027172
450	0.12502323	215.6654882	4.904302	5702.251936	0.000579	0.02485885	0.02485885	0.036443
500	0.15058964	270.2211709	4.768422	5506.638687	0.000557	0.0306138	0.0306138	0.046776
550	0.18090612	330.8344066	4.749365	5420.108244	0.000547	0.03669294	0.03669294	0.057527
600	0.21646488	405.4654389	4.738812	5350.976885	0.000534	0.04368371	0.04368371	0.070437
650	0.25453544	495.0548997	4.705337	5280.499774	0.000514	0.05131895	0.05131895	0.085716



Mean Average Precision : 0.000695

2.2. Method 2: Collaborative-based filtering

Workflow:



2.2.1 Data Preparation

For data Preparation we have considered 3 tables [User , Reviews , Business]
In Task 1 we have generated User - Business. This data contains information about User and all the businesses visited by that user in North Carolina.
With the help of data generated in Task 1 we have created a new table which consists of 3 columns User,Business Visited by the user and the rating of the business.

Sample User ID , Business and Rating Data :

User ID	Business ID	Rating
cZqfqgMg7yUJHnAmYbb0FA	jYqOPpSmtKbKzf0Z_g-Oyg	3.5
uGZgPyXMQ2cXFqho2Sm0FA	58Zb67Gl0X1IGNCAYKbrIA	4

Dimensions of the above matrix : 140595 x 3

With the help of above generated table we have generated a sparse rating matrix in **R,** with **User ID** as rows and **Business ID** as columns. Which looks very much similar to matrix as shown below.

	Business 1	Business 2	Business 3
User 1	3.5	-	-
User 2	2.5	-	4

2.2.2 Modelling and Training

We have modeled our data using **User Based Collaborative Filtering** with a memory based technique for training on the available data.Usually the memory based technique uses the rating data of the user to compute the similarity between user or items which is used for recommendations. In this approach value of ratings of user 'u' given to business 'i' is calculated as an aggregation of some similar users rating of the business.

$$r_{u,i} = \text{aggr}_{u' \in U} r_{u',i}$$

Where **'U'** denotes the set of top **'N'** users that are most similar to user **'u'** who rated business **'i'.**

In the above functions k is the normalizing factor and is defined as mentioned below.

$$k = 1/\sum_{u' \in U} |\text{simil}(u, u')|,$$

For this problem we have used the cosine based approach in which we will use cosine similarity between 2 users.We have used top-N- Recommendation algorithm using the similarity based vector model to identify k most similar user and after finding the users we will be aggregating the corresponding records of that particular users to identify the set of items to be recommended. Below is the formula for **cosine similarity.**

$$\text{simil}(x, y) = \cos(\vec{x}, \vec{y}) = \frac{\vec{x} \cdot \vec{y}}{||\vec{x}|| \times ||\vec{y}||}$$

We have used **80%** of the data for training and remaining **20%** of the data for **Testing and Evaluation** purpose.

To achieve this following task we have used **recommenderlab** package in **R.** Using **recommenderlab** we have generated the **sparse rating matrix** for the given data and trained the sparse matrix using User based collaborative filtering using **cosine similarity.**

2.2.3 Testing

For Testing we have used used the above training dataset to predict n recommendations for the users.
Here are some of the sample top-10 recommendations we have generated for two users.

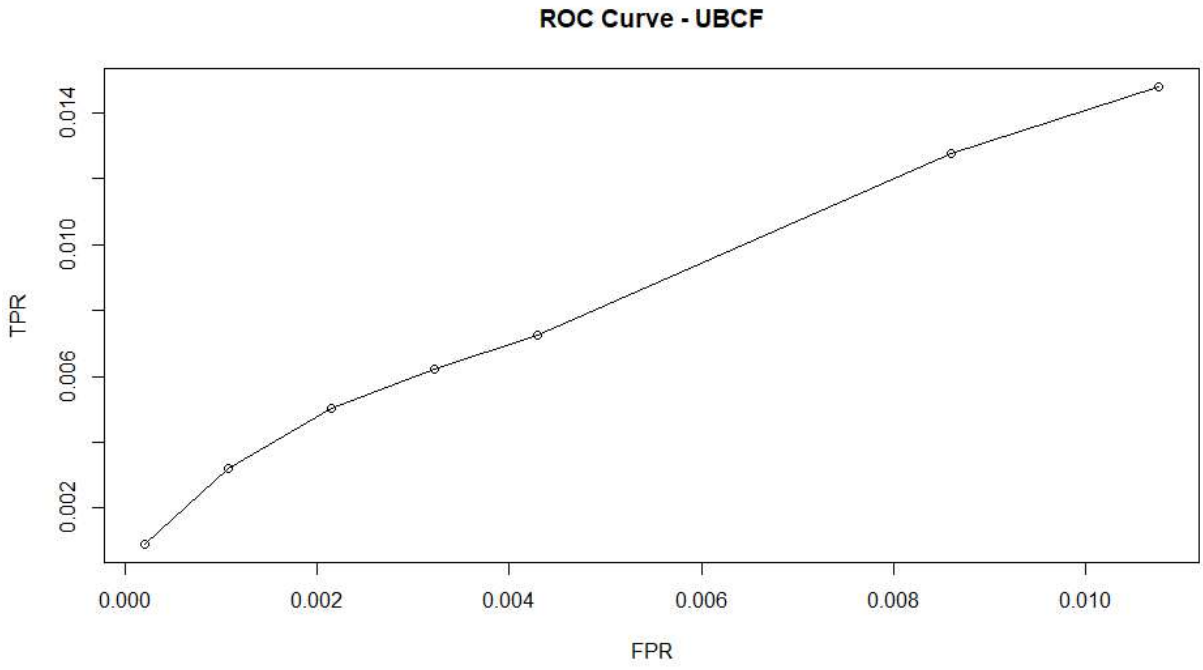
User --->		**User ID [--3WaS23LcIXtxyFULJHTA]**	**User ID [-DmrFVzkT3bkDcmeq1tbtQ]**
Business ID Recommendations	**1**	hggVnGwA5042-ABxqeJX-A	--cZ6Hhc9F7VkKXxHMVZSQ
	2	5q6Xh-UcJa78bp6dzyaE7w	--KCl2FvVQpvjzmZSPyviA
	3	5q6Xh-UcJa78bp6dzyaE7w	-_nz_8EPGQKKTd8loQDgXQ
	4	22MYhTXwSXaS4rW2VOrR-w	-_Ph_Y9mJWNdKqArDxp0lQ
	5	R1jJQi2yR44D_2ileqr8kA	-0QtTRrAMn6DKLZNef3Ojg
	6	vqG1Z2XpS_PryPsfY0CSng	-2pmn-oTJeybmDrL-ojwrw
	7	6E0D-5wVjDJZQF8iA6k3ig	-2pQf1ceDZyE2ReCNbj-3A
	8	e9sB72njxz87r5TL6kG	-5L8zOxibac-vBrsYtxXbQ
	9	WPV0ucYjnb2HOrNxSopv6A	-5XuRAfrjEiMN77J4gMQZQ
	10	bgvm73MMjC2f5qo_RgXOXg	-7VzJ1aG5yuWB9LT42yhlw

2.2.4 Evaluation

2.2.4.1 Confusion matrix for few queries

User order [sparse matrix]	TP	FP	FN	TN	precision	recall	TPR	FPR
1	0.00545	0.440588	2.049767	2190.504	0.012219	0.001049	0.001049	0.000201
3	0.013123	1.324991	2.042094	2189.62	0.009807	0.002381	0.002381	0.000605
5	0.018573	2.211617	2.036644	2188.733	0.008328	0.003222	0.003222	0.00101
10	0.032126	4.428254	2.023091	2186.517	0.007203	0.005509	0.005509	0.002022
15	0.043958	6.646612	2.011259	2184.298	0.00657	0.007398	0.007398	0.003035
20	0.05407	8.866691	2.001147	2182.078	0.006061	0.009012	0.009012	0.004049
40	0.081606	17.75991	1.973611	2173.185	0.004574	0.012834	0.012834	0.008111
50	0.089494	22.21241	1.965722	2168.732	0.004013	0.014455	0.014455	0.010144
60	0.095662	26.66662	1.959555	2164.278	0.003574	0.01549	0.01549	0.012179
100	0.113589	44.49021	1.941628	2146.455	0.002547	0.020057	0.020057	0.020319
150	0.13711	66.76859	1.918107	2124.176	0.002049	0.026507	0.026507	0.030494
250	0.179993	111.3295	1.875224	2079.615	0.001614	0.03733	0.03733	0.050846
300	0.199498	133.6119	1.855719	2057.333	0.001491	0.042177	0.042177	0.061023
350	0.216995	155.8963	1.838222	2035.048	0.00139	0.046534	0.046534	0.071201
400	0.23858	178.1766	1.816637	2012.768	0.001337	0.052743	0.052743	0.081377
450	0.276085	200.441	1.779132	1990.504	0.001375	0.064263	0.064263	0.091545
500	0.31488	222.7041	1.740337	1968.241	0.001412	0.073604	0.073604	0.101713
550	0.354464	244.9664	1.700753	1945.978	0.001445	0.08441	0.08441	0.111881

2.2.4.2 ROC curve - UBCF



Mean Average Precision : 0.004128

For model comparison we can use Area under the ROC curve for effectively comparing the two methods we used in Task 1.

For a recommendation problem especially for the large data sets the **MAP** values are always **pretty low**, Though MAP is roughly the average area under precision recall curve.

We feel like for a model comparison especially between Content based filtering and Collaborative Filtering Area under ROC curve gives a good representation of the two models.

2.3 Comparing Method 1 and Method 2 for Task 1

Mean Average Precision For Task 1 Model 1 : 0.000695

Mean Average Precision For Task 1 Model 2: 0.004128

Mean Average Precision for Model 1 is high when compared to the Model 2.

3. TASK 2: Predicting most famous dishes in a restaurant

3.1 Research Question

Our main idea is to predict the most famous dishes in a particular restaurant and display it to the user. For this we are using reviews and tips to extract this information. Reviews and tips include many useful information what are the dishes that user has tried in that restaurant and their feedback about it.

From reviews and tips, we can say that user has liked a dish or user disliked a dish because it wasn't that great. Our main idea is to capture this most information and identify the top dishes and display it to the user.

Usefulness to yelp:

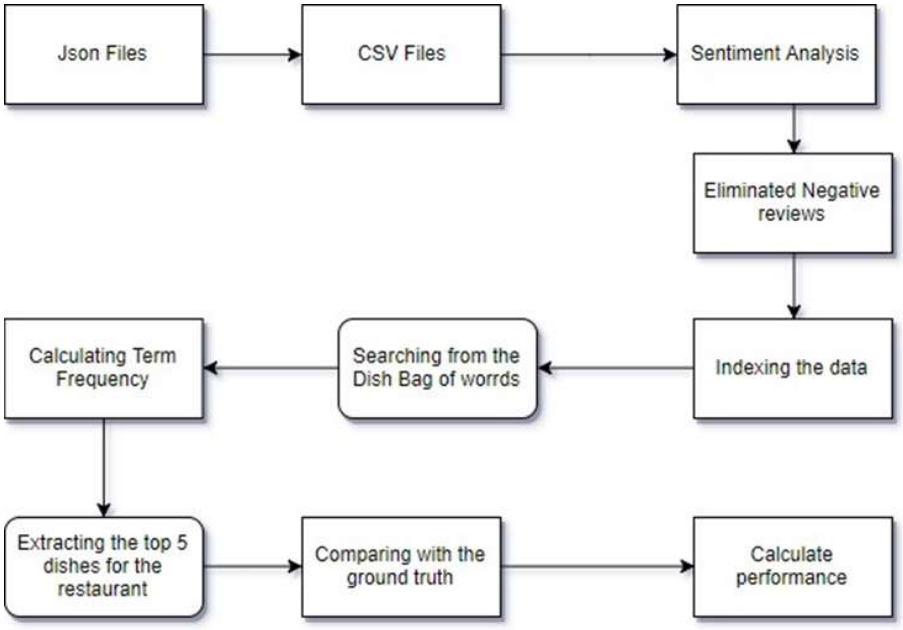
Many users have to manually read through all the reviews and tips to know what dishes are good in that particular restaurant.

We save user's hassle of reading all the reviews and tips manually as it is time consuming and maybe futile approach based on the total number of reviews. To eliminate this hectic process, we are predicting the famous dishes of a particular business by analyzing the reviews and tips.
This is a crowd sourced approach as we are using all the data available in the reviews and tips based on the content written by the user.

Data Extraction:

Due to the large size of the dataset, we have restricted to popular Mexican dishes and restaurants across USA serving this cuisine. We extracted all the reviews and tips of all the mexican restaurants and also some other important attributes like Business ID, User ID, Review ID, Rating stars, Sentiment analysis score and Review text.
Total number of reviews and tips of all the mexican restaurants: 60580

Flow Diagram:



Data Preprocessing:

This section basically explains how the data is preprocessed and here we are using sentiment analysis approach to pre process data.

Sentiment analysis:

Sentiment analysis helps us in understanding the tone of the person in a text. Whether it is a positive tone or negative can be identified using packages like afinn in python. This afinn already has a list of negative words and positive words. With this it calculated a P score for each review and tip.

If the pscore > 0 - then the review is positive
If the pscore = 0 - then the review is neutral
Is the pscore <0 - then the review is negative

In our scenario, we extracted all the positive and neutral reviews.
Why sentiment analysis?
As we are concentrating on the top 5 dishes, we concentrated only on the positive and neutral reviews as they hold that information.

Method:

Extracted reviews and tips for each business serving the mexican cuisine.
Performed sentiment analysis on the available text to generate a score which will help us remove all the reviews or tips which have a negative score.
Not only based on the review rating as there exists situation where people like the food but not the service or ambience.

Eg: "*I ordered carne asada, fish, & chicken tacos. Tacos are really small & over cooked. Was really disappointed, expected a lot based on the reviews I read.You're better off going to a taco truck. I really hate how overpriced tacos are now. Wouldn't mind paying $2-3, but at least give me what I'm paying for*".

Total no. of reviews and tips before filtering: 60827
Total no. of reviews and tips after filtering: 54486

Index generation:
Indexed the data using Lucene indexing and then searched across the documents by building query. Business ID, User ID, Review ID, Rating stars, Sentiment analysis score, Review text parameters are indexed.

Query Generation:
We first extract the unique businesses of the dataset using Terms from the Lucene dataset.
The general QueryParser for dishes which just have a single word in it.
For dishes with more than 1 word, we have used Phrase QueryBuilder.
Then, we use Boolean query to combine the results and get the best dishes for all the businesses.

Predicting the best Mexican dish
Manually created a Mexican dish list from many sources available on the internet.
Eliminated the business which have reviews less than 50, due to lack of enough data to compute the top dishes.
Obtained the number of occurrence of each dish in the dish list and obtained a score.

Search Approach:
We used two approaches to search for the dishes using the dish list.

Exact Match:
Here, we try to match the exact words as present in the dish bag of words which may or may not be present in the review text.

Partial Match:
On a set of data, we extracted the dish name and applied stemming algorithm to get the name of dishes.This ensured that different variants including plural forms as well as singular forms are considered as a single entity.

Ranking the data:
Sorted the dishes basing on the score for a particular business as per the term frequency score. This will be used to retrieve all the top 5 dishes of the restaurants.
Eliminated all the dishes that have a score of 1 as a single occurrence is not useful enough.

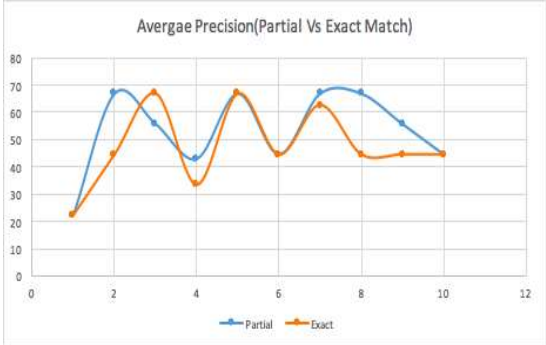
3. Evaluation Technique
Since, this is an Unsupervised learning approach it is difficult to evaluate the results as there are no labelled train data sets available. We have created a manually filtered result of top dishes for 50 restaurants to compare the performance of the algorithm.

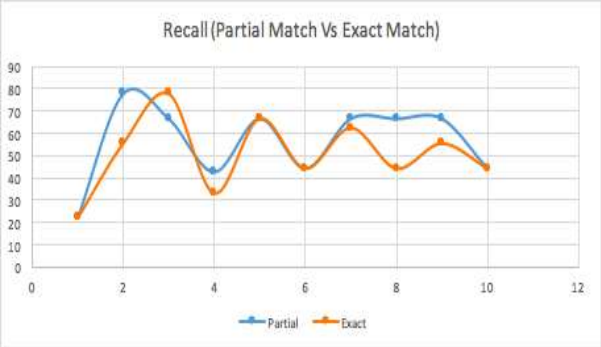
3.1 Ground Truth Generation

- To generate ground truth, we manually filtered the dishes out of 50 restaurants serving Mexican dishes.
- We checked the tone of the review to manually estimate the sentiments and filter out all those which seemed negative.
- We did POS Tagging and extracted the noun phrases from the review text and also stemmed the words.Thus, we received top dishes for 50 Mexican restaurants.

We have also generated word clouds, to understand what are the top dishes in a restaurant. Below is a sample word cloud.

For evaluation we used recall and Map to calculate the accuracy. Below is the comparison of the performance between partial and Exact match for Recall and Average Precision.





The tables used to generate these graphs are in the github and also the MAP of Partial Match was 57.365086 and Exact Match is 53.56945.

4. Conclusion

Challenges:

1. The limitations of using a fixed bag of dishes list is that some restaurants have innovative dish names which might be different from their original dish names.
2. The prediction system fails to identify them and thus the dishes gets eliminated in spite of being popular enough to be classified it as best dish for that restaurant.

Future Scope:

1. We can expand the bag of words set to include different cuisines so that we can support more and more restaurants on yelp and help users get information about the best dishes of those restaurants.
2. We can use Named-Entity Recognition to actually filter the dishes out instead of using the dish bag of words.

Conclusion

The Yelp dataset was used for 2 tasks. In the first Task approach 1, we have recommended restaurants using the user's previous ratings by creating a model user profile and categorize the user based on the price range and the cuisine served at those restaurants and for the same question approach 2 we have used User,Business and average business rating of the user to recommend the businesses to the user, Out of the two approaches, approach 1 using content based filtering turned out to be good. For the second task, we predicted the top mexican dishes of the restaurants serving the mexican cuisine. This was done by pre-processing the data using sentiment analysis and applying filtering techniques like stemming, stop words removal, parts of speech tagging, word cloud generation. Later, Lucene query generation was used to extract the partial match and full match of the dishes from the dish list.