Recommendations from Yelp dataset

Discovering attributes and connoisseurs

Surya Mani Deepak Kancharla
University of Colorado, Boulder
Surya.Kancharla@colorado.edu

Koushik Chennugari
University of Colorado, Boulder
Koushik.Chennugari@colorado.edu

Laxmi Naga Santosh Attaluri
University of Colorado, Boulder
Laxmi.Attaluri@colorado.edu

Rajath Tellapuram
University of Colorado, Boulder
rajath.tellapuram@colorado.edu

ABSTRACT

The aim of the project is to extract meaningful insights from the yelp data-set which can be of business value. In this project, we will be extracting the most liked and disliked attributes of a restaurant by mining useful information from yelp data-set. Attributes can be food items, cuisine, ambience, pricing, etc. Especially finding liked and disliked attributes by glancing through all the reviews for a given restaurant can be very difficult. We plan to build a system which extracts attributes from reviews and attach sentiment to those attributes. Additionally, it is difficult for people to know which reviews are actually helpful and which are fake or not useful. We intend to solve this problem by extracting the connoisseurs related to the queried topic and present these connoisseurs to the user as the reviews given by them reliable and most valuable. The connoisseurs of a given area are extracted using the information from yelp data-set.

KEYWORDS

Topic Modelling, Natural Language Processing, Sentiment Analysis, Python, Clustering

ACM Reference Format:
Surya Mani Deepak Kancharla, Koushik Chennugari, Laxmi Naga Santosh Attaluri, and Rajath Tellapuram. 2018. Recommendations from Yelp dataset: Discovering attributes and connoisseurs . In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Yelp is the go-to data-set for people to get restaurant recommendations. They have a lot of information regarding the restaurants, reviewers etc. Our objective is to identify the most liked and disliked items of a restaurant and to come up with a way to find the connoisseur to follow in a specific location. We refer to them as local connoisseur in the rest of the paper, essentially they are reviewers who are experts on a particular category within a certain location. This makes the local connoisseurs a credible source of information when a person wants to know about whose reviews to consider about the restaurants given a location. We intend to build a system using natural language processing and machine learning techniques which can extract the key attributes of a restaurant like food quality, service etc from the reviews and find the sentiment attached to the them. This helps us in scoring the key attributes attributes of the restaurant. We use classifiers to find out the experts of a category, which may include using decision trees etc., further the experts are localized based on geographical locations using clustering techniques.

2 MOTIVATION

Generally, most of the people use yelp for restaurant recommendations. But, we get very little insight about the restaurant when we go through the rating, price range, reviews, etc. It is not possible to scan through all reviews for a user as it is a tedious job due to the huge numbers and lack of credibility in them. This has motivated us to build, an application that provides, for each restaurant rating of the key attributes, which can help the user get a good idea about the restaurant. Also, due to the ease in access granted to anonymous people in publishing comments or reviews about a particular topic, it becomes really difficult to know which reviews are actually credible and which of them are not. Thus, we intend to build a system which extracts expert reviewers from all set of reviewers for a particular locality, so that a person could follow people who give credible and valuable information.

3 RELATED WORK

We were motivated to choose our problem statement based on the concepts and approaches from the following papers.

The paper [4] introduces an unsupervised learning technique which identifies product features from the reviews. Furthermore, identifies the opinion and its polarity for those features.

In this paper [2], the goal was to use several filtering models to accurately identify food mentions and match them up with menu items. This process would extract the key terms from the reviews and try to find a gold match by annotating the data. For the annotations a crowd sourced annotation tool is used. Further an sentiment analysis model is used to categorize these mentions and give a polarity score to them, which would help in classifying the dislike or liked items. The way this paper achieves is by using Stanford

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

CoreNLP library to extract mentions from the reviews, this returns all mentions found within reviews, even non-food items. Then they are filtered so only food mentions are left, using named-entity recognition. After these food mentions are extracted, a human input (via the annotation tool) is used to figure out the sentiment and determine if a dish is positively, neutrally, or negatively reviewed. Finally, an aggregate of the sentiment is taken from all the reviews, for each menu item analyzed.
The above two papers motivated us to take up our problem statement of finding and rating the key attributes of a restaurant.We will find an approach to find the attribute related to a review and the polarity of the attributes can be extracted using sentiment analysis.

In this github project [1], the user tried to extract the food items from reviews using NER (Named Entity Recognition). We tried this approach initially to extract the key attributes of a restaurant.

The paper[3] identifies the local experts for Twitter data-set by proposing a local expert framework which combines topical authority and local authority. We plan on implementing topical authority using classifier. Similarly, we will implement local authority using clustering technique.

In this paper[6], a method of classifying domain expertise on hierarchy of categories is presented. The geo-spatial datasets that were used mainly were Twitter and Foursquare, this data was used as the most frequent places of communication or taking advice was the online websites where people get a chance to connect to their friends or family to obtain advice. This paper has different metrics to tag a person as a domain expert, mainly two methods are effective, those are, Within-class coverage Within-class diversity. Here class can be taken to be analogous to a domain, first, Within-class coverage metric is about how frequently the person interacts with that domain, example, how frequently a person visits places regarding food. Second, Within-class diversity is a metric about how diverse is the person related to a domain subject, example, how diverse is the person regarding the domain of food, which may include multiple experiences with a vast array of cuisines. Thus, this was the approach taken to establish domain expertise, this paper is still an ongoing piece of work in progress.

4 PROPOSED WORK

4.1 Datasets

We plan on using yelp dataset from kaggle.
business.json - Gives us details about restaurants, particularly the fields used are:

1. ”business_id”: To identify each restaurant uniquely
2. For Location: ”city”,”state”,”latitude”,”longitude”
3. ”is_open”: To check if the restaurant is still operating
4. ”categories”: Offered cuisines at the restaurant

Sample business.json of the required fields

```
{
”business_id”: ”tnhfDv5Il8EaGSXZGiuQGg”
”city”: ”San Francisco”
”state”: ”CA”
”latitude”: 37.7817529521
”longitude”: -122.39612197
”is_open”: 1
”categories”: [”Mexican”,”Burgers”,”Gastropubs”]
}
```

review.json - Gives us details about reviews given to restaurants, particularly the fields used are:

1. ”review_id”- Unique ID to identify each review
2. ”user_id” - Unique ID to identify each user who has given a review
3. ”business_id”- Unique ID to identify each restaurant to which the review is given
4. ”stars”-Review’s rating
5. ”date”-The date when the review was given
6. ”text”- The actual review text

Sample JSON:

```
{
”review_id”: ”zdSx_SD6obEhz9VrW9uAWA”
”user_id”: ”Ha3iJu77CxlrFm-vQRs_8g”
”business_id”: ”tnhfDv5Il8EaGSXZGiuQGg”
”stars”: 4
”date”: ”2016-03-09”
”text”: ”Great place to hang out after work: the prices are
decent, and the ambience is fun. It’s a bit loud, but very lively.
The staff is friendly, and the food is good. They have a good
selection of drinks.”
}
```

newline user.json - Gives us details about the user, particularly the fields used are:

1. ”user_id”: Uniquely identifies each reviewer
2. ”name”:Name of the reviewer
3. ”review_count”:Number of reviews given
4. ”elite”:Classifies the reviewer to be an elite member for certain years
5. ”average_stars”:Average rating of the user

Sample user.json:

```
{
”user_id”: ”Ha3iJu77CxlrFm-vQRs_8g”
”name”: ”Sebastien”
”review_count”: 56
”elite”: [ 2012, 2013 ]
”average_stars”: 4.31
}
```

tip.json-Gives us details about the comments made on the restaurants, particularly the fields used are:

1. ”text”: ”Secret menu - fried chicken sando is da bombbbbbb Their zapatos are good too.”,
2. ”date”: ”2013-09-20”,
3. ”likes”: 172,
4. ”business_id”: ”tnhfDv5Il8EaGSXZGiuQGg”,
5. ”user_id”: ”49JhAJh8vSQ-vM4Aourl0g”

Sample tip.json

```
{
”text”: ”Secret menu - fried chicken sando is da bombbbbbb
Their zapatos are good too.”,
```

”date”: ”2013-09-20”,
”likes”: 172,
”business_id”: ”tnhfDv5Il8EaGSXZGiuQGg”,
”user_id”: ”49JhAJh8vSQ-vM4Aourl0g”
}

4.2 Tasks

4.2.1 Discovering Connoisseur. There are multiple stages involved for identifying local connoisseur in a given locality. Based on the paper [3] ,we can basically divide the problem into two sub-problems. Firstly, we need to find the list of connoisseur for restaurants based on yelp data-set and finally filtering out the connoisseur who are in the same locality of that of the query.

(1) We train a model for classifying the users as expert or not. This task involves devising different features such as number of reviews, yelp age, average review length, variance in rating, etc. We plan on using different classifiers such as SVM, Random Forest and compare the classifiers based on the accuracy obtained.
(2) In order to find the users in a particular locality, we need to identify the location of the user but unfortunately that information is not available in the data-set. The data-set consists of business location (coordinates). We use this information for approximating the connoisseur user’s location by extracting the locations of the businesses user has ever reviewed. Based on this generated data, we use a GMM(Gaussian Mixture Model) clustering algorithm to identify the clusters where the user has an expertise. Eventually, we gauge the distance between the query location and the centers of the user’s clusters. If this distance is less than or equal to the threshold value, then we classify the user as local connoisseur.

In summary, first we filter the expert users from the list of users using the machine learning classifiers. Then we find the locality of there expert users and filter the users whose locality resides in the same vicinity as that of query location.

4.2.2 Rating key attributes of a restaurant. Looking through restaurant reviews on Yelp to find what is good about a restaurant is difficult. So, we wish to rate the key attributes like Food Quality, Service, Ambiance and Value For Money for a given restaurant using the reviews data-set. Since there is no labeled dataset, we will first start experimenting different approaches to extract restaurant attributes for a given review.

Once the attributes are extracted, we apply sentiment analysis on the attributes to know if it was mentioned in a positive way or negative. We do this for all the reviews for the restaurant. Then we will average them and display the rating s of the mentioned attributes for all the restaurants.

5 IMPLEMENTATION

5.1 Data pre-processing

5.1.1 Data Cleaning. Yelp dataset covers a lot of categories such as restaurants, automotives, bar, shopping etc. We have cleaned it to focus only on the restaurants related category. We have used MongoDB for storing the raw json data. The Yelp dataset is sparse i.e., many values are missing in the dataset. We had to deal with sparse data by filling the missing values by average values. We removed the noisy data.For example, rating for a review was 05-04-2016 which is totally irrelevant. We have completely ignored such data.

5.1.2 Data Integration. We also had to join different files to get the data in the desired format. We have different datasets related to Yelp such as business data,user data and reviews data. We had to combine these files and do pre-processing.

5.1.3 Data Transformation. The dataset was huge and required lot of preprocessing, so we have preprocessed the data using spark by launching an EMR cluster in AWS. We had to do lot of data transformations. Few of them are :

(1) We had an attribute called “yelping_since” which basically tells us about the year in which the user has created an account . We had to convert that to age to use it as a feature for training the model.
(2) We had to derive attributes like average and standard deviation of ratings given by the user to all the restaurants

5.2 Topical authority

We are trying to find whether the user is knowledgeable for the “restaurants” category We have considered following features which would be helpful for determining whether a user is knowledgeable or not

(1) Total number of reviews given by the user in “Restaurants” category
(2) Average rating given by the user in “Restaurants” category
(3) Standard deviation in ratings
(4) Number of useful votes user got in “Restaurants” category
(5) Number of years since user created the account
(6) Number of unique restaurants reviewed by the user
(7) Number of friends for the user

Based on the above features, we have trained our model using Random Forest classifier and Support Vector Machine(SVM).
Apart from the features that are mentioned above, we have thought of few more features which would have improved the accuracy and F-measure:

(1) Average length of user’s review in the “restaurant” category
(2) Total reviews given by the user
(3) Number of cool votes user got in “Restaurant” category
(4) Number of funny votes user got in “Restaurant” category
(5) Average rating given by the user over all the categories

However, adding those attributes did not make any considerable difference in the accuracy or F-Measure of the classifiers.

5.3 Location authority

Yelp data set does not have user’s location information. This has become one of the biggest challenge for making accurate predictions. Hence, we have used unsupervised learning i.e., Gaussian Mixture Model clustering to determine the location of the user. We are using the business location i.e., restaurant location which were reviewed by the user to determine the location of the user. We have used Gaussian Mixture Model(GMM) to form the cluster based on the restaurant's coordinates for every elite user. We have considered the centroid of the most dense cluster as the location of the user. We are using 25 miles as the minimum threshold distance based on which we classify whether the predicted elite users are local or not.

5.4 Rating key attributes of a restaurant

We are trying to evaluate attributes of a restaurant. We have decided on few aspects which we will be evaluating like

(1) Food quality
(2) Ambience
(3) Service
(4) Value For Money

There is no labeled dataset for the problem we are trying to solve. So, to generate the labels we used topic modelling. As a first step we preprocessed the review in 4 steps.

(1) Splitting the review into sentences and the sentences into words.
(2) Lowercase the words and remove punctuation.
(3) Stop-words and words that have fewer than 3 characters are removed.
(4) Words are lemmatized, i.e. words in third person are changed to first person and verbs in past and future tenses are changed into present.
(5) Words are stemmed, i.e âĂŁwords are reduced to their root form. by removing stop words.

For topic modelling we tried two approaches, LDA (Latent Dirichlet Allocation) [7] using Bag of Words and LDA using TF-IDF [5]. We implemented it on the reviews to cluster them into topics. We performed the topic modelling with number of topics as 5 and 10. Then, we assigned the topics generated to the above mentioned attributes.

Once the model is ready, we save it to the system which makes it easy to use multiple times. Then, we implemented the attribute rating module where we maintain a map of restaurants. For each entry in the map, business-id is the key and value is a map of attribute's ratings. In this module we iterate over the reviews and for each review, we perform the following three steps

(1) Split the review into sentences.
(2) For each review find the topic it belongs to and find the attribute related to that topic.
(3) Then using a sentiment analysis tool find the sentiment level attached and update the values in the map accordingly.

We have used StanfordCoreNLP to perform the sentiment analysis. The library takes a text as parameter and returns different values like sentiment and sentiment level. Sentiment has values neutral, negative and positive. Sentiment value is an integer ranging from 0-4 (Very negative to very positive).

So, we run the above steps for the whole review dataset. After that, for each restaurant we calculate the average ratings for the four attributes mentioned and write it to a file.

6 EVALUATION

(1) Using attribute called stars from the dataset, we computed average rating of each restaurant. Then, we averaged the ratings of all the attributes computed and used the difference between actual average rating and computed average rating of the restaurants as the evaluation metric.
(2) Yelp tags few users as elite members on the basis of whether the user is active and influential in a particular vicinity. We would use these list of elite members as benchmark for the evaluation. All the users who have been a elite member for at least a year can be categorized as positive example whereas rest of the members as negative example.

7 TEAM MEMBERS CONTRIBUTION

(1) Surya Mani Deepak Kancharla: I worked on the second problem statement, Rating the key attributes of a restaurant. I worked on the data preprocessing of this problem and initially implemented a CRF (Conditional Random Field) model with custom features to perform Named Entity Recognition. Later, worked on topic modelling of the review text and finding topics related to the attributes. I worked with Koushik on building the LDA model. Then, I wrote a module for attribute rating where initially for each review we identified the attribute it belongs to using the LDA model. In the next step, I used the StanfordCoreNLP library to find the sentiment level. Finally, averaged the attribute scores for each restaurant and stored it in a file.
(2) Laxmi Naga Santosh Attaluri: I worked on the first problem statement i.e., Finding Local Connoisseur. I worked on data preprocessing which involves data cleaning and data transformation. Later I worked on feature engineering and random forest model for topical authority along with Koushik. I have worked on SVM model along with Rajath as part of finding the connoisseur (Topical Authority). Worked to find the local connoisseur (Location Authority) by clustering the connoisseur to find the location of those experts where we have used the Gaussian Mixture Model. After clustering the data to find the connoisseur's location, I worked along with Rajath towards finding the closest local connoisseur given the input location.
(3) Rajath Tellapuram : Worked on the first problem statement, where we had to find the local connoisseurs. Worked on the following with Santosh, initially started with data-preprocessing, where we had to clean and transform sparse data, noisy data. Implemented the Support Vector Machine model and figured out the features required for the model to evaluate if the user is a connoisseur or not which is the Topical Authority Model. Worked on clustering these connoisseur using the Gaussian Mixture Model, with Santosh, where we could find connoisseur in that locality of the given input of a location.
(4) Koushik Reddy Chennugari : I worked on parts of both first and second problem statement. Initially we tried to come up with a set of features to build the model for topical authority. Once we finalized on features, I worked on building random forest model for predicting people who have topical authority along with Santosh. I also worked on building LDA model to decide the aspects for which rating will be calculated for all the restaurants along with Deepak. There were two approaches for the building LDA model which are bag of

words model and TF-IDF model. Eventually topic modelling is done using bag of words model as it produced better topics.

8 DISCUSSION

8.1 Local Connoisseur

The initial obstacle was to find an evaluation metrics. After discussion with our Professor, we were able to consider elite member attribute present in the user dataset for determining if the user is local connoisseur or not.
We also did not have any user location to calculate the locality of the user. After brainstorming and discussing with the Professor, we were able to discover a way to find the approximated location of the user.

8.2 Rating key attributes of a restaurant

The second problem statement we are solving, rating the key attributes of a restaurant, does not have a labeled dataset. so, initially we worked on implementing a NER (Named Entity Recognition) on the review text to extract the entities such as food items. Once those were extracted, we planned on performing a sentiment analysis to get the sentiment attached to the entity. But, this approach was unsuccessful as the standard NER packages like NLTK and Stanford-CoreNLP were not able to accurately predict the entities we wanted. For example, a food item was getting assigned organization, person etc. So, I implemented a CRF (Conditional Random Field) model with custom features to perform the named entity recognition. But this model also didn't perform to our expectations.

After that, we planned on selecting a sample size of 10,000 reviews from the dataset which correctly represents the whole dataset and manually label each sentence of a review with an attribute. But, this approach seemed in-feasible.

Later, we discussed our problem with the professor who directed us towards topic modelling. We read about Latent Dirichlet Allocation and understood how it works. Then we implemented LDA module using tf-idf and bag of words approach and tried different number of topics like 5 and 10. Then, mapped the generated topics to key attributes. This way, we were able to generate a labeled dataset.

9 RESULTS

9.1 Topical Authority

We have trained the model using Random Forest and Support Vector Machine.

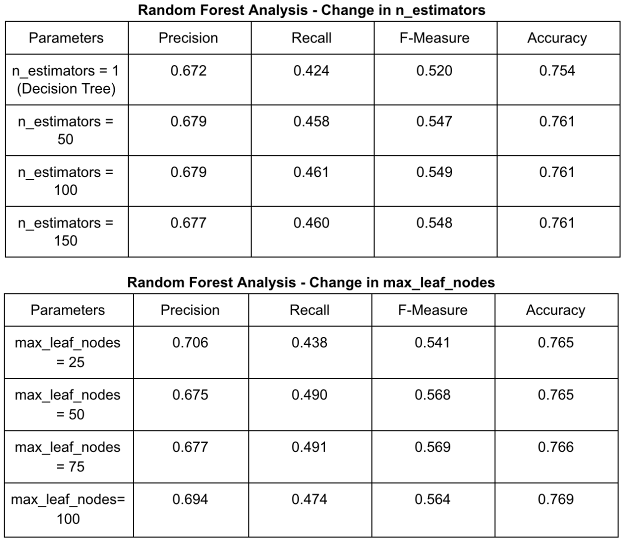
9.1.1 Random Forest. We have used 5-fold cross validation and were able to obtain an accuracy of 0.92.
However, the dataset is very skewed i.e., the positive examples are very less when compared to negative examples. Hence we cannot consider accuracy as correct measure of correctness. For skewed data , F-measure would be a much appropriate measure for determining the correctness of the results.

$$F - measure = \frac{2 * (precision * recall)}{(precision + recall)}$$

We got the following results:

(1) precision=0.46
(2) recall=0.08
(3) F-measure = 0.14

The above results were obtained when we have used the complete dataset. The recall value obtained based on the above mentioned features is very less.So, we have trained the model using a subset of dataset which represents the complete data set. We have reduced the skewness considerably in the subset which we have chosen.



Random Forest Analysis - Change in n_estimators

Parameters	Precision	Recall	F-Measure	Accuracy
n_estimators = 1 (Decision Tree)	0.672	0.424	0.520	0.754
n_estimators = 50	0.679	0.458	0.547	0.761
n_estimators = 100	0.679	0.461	0.549	0.761
n_estimators = 150	0.677	0.460	0.548	0.761

Random Forest Analysis - Change in max_leaf_nodes

Parameters	Precision	Recall	F-Measure	Accuracy
max_leaf_nodes = 25	0.706	0.438	0.541	0.765
max_leaf_nodes = 50	0.675	0.490	0.568	0.765
max_leaf_nodes = 75	0.677	0.491	0.569	0.766
max_leaf_nodes= 100	0.694	0.474	0.564	0.769

Figure 1: Random Forest Evaluation and Results

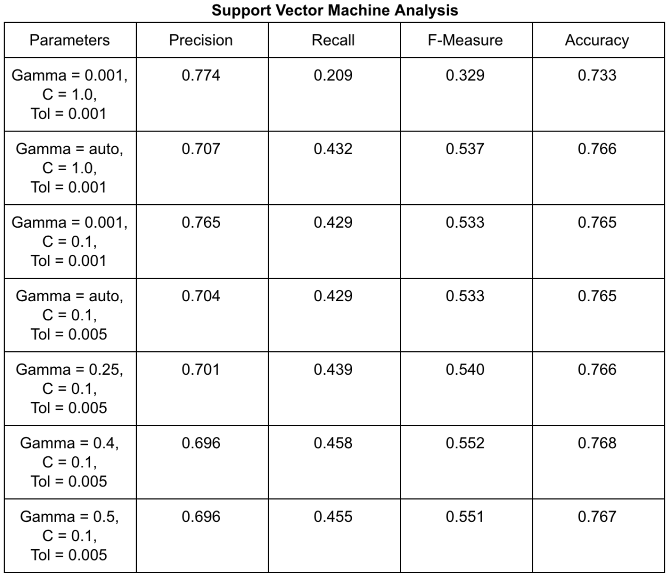
Figure 1 depicts the change in the F-Measure and accuracy as the number of support vector machines (n_estimators) are varied. Also when n_estimator is set to 150, the second table in Figure 1 shows the change in accuracy and F-measure as max_leaf_node (Maximum number of leaf nodes in the trees grown). The best results that we were able to get was:

(1) precision=0.69
(2) recall=0.47
(3) F-measure = 0.564
(4) Accuracy = 0.769

9.1.2 SVM. The accuracy and F-measure obtained using SVM are more or less similar to the the accuracy and F-measure obtained using RandomForest. Figure 2 depicts the F-Measure and Accuracy for different tuning parameters. We parameters which we have tried tuning are gamma (Kernel Coefficient), C (penalty parameter) and tol (tolerance for stopping criterion).

9.2 Location Authority

As we have used Gaussian Mixture Model, we clustered each of the user's rated restaurants into two clusters. After clustering them, we choose the the cluster which is more dense and find the centroid of it. As we can see in Figure 3, they are two figures of different users, in the first figure, the density of the clusters are 45 and 57, the centroid is 35.2250973639,-80.8411199818 which is the green



Support Vector Machine Analysis

Parameters	Precision	Recall	F-Measure	Accuracy

Gamma = 0.001,
C = 1.0,
Tol = 0.001
	0.774	0.209	0.329	0.733

Gamma = auto,
C = 1.0,
Tol = 0.001
	0.707	0.432	0.537	0.766

Gamma = 0.001,
C = 0.1,
Tol = 0.001
	0.765	0.429	0.533	0.765

Gamma = auto,
C = 0.1,
Tol = 0.005
	0.704	0.429	0.533	0.765

Gamma = 0.25,
C = 0.1,
Tol = 0.005
	0.701	0.439	0.540	0.766

Gamma = 0.4,
C = 0.1,
Tol = 0.005
	0.696	0.458	0.552	0.768

Gamma = 0.5,
C = 0.1,
Tol = 0.005
	0.696	0.455	0.551	0.767

Figure 2: Support Vector Machine Results

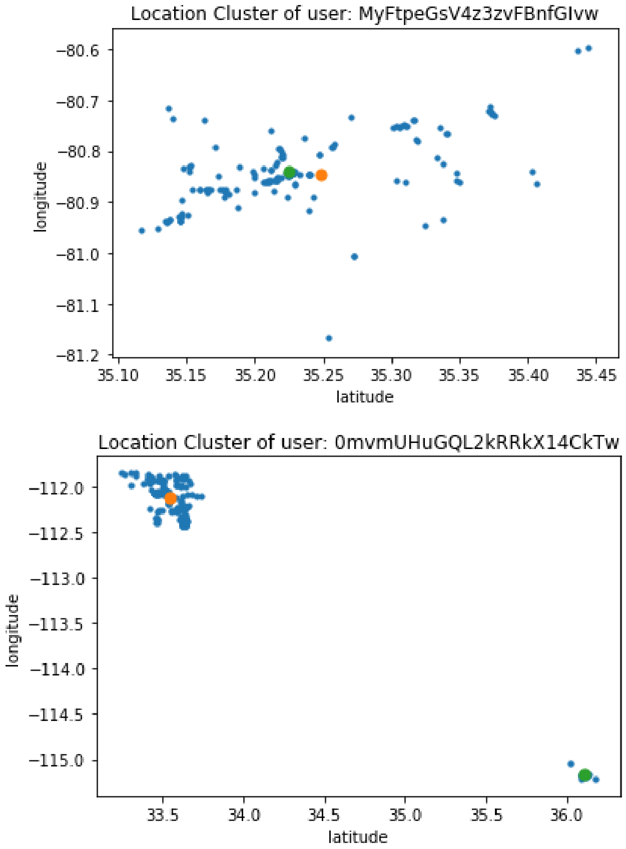


Figure 3: Gaussian Mixture cluster

colored point. Similarly for the second figure, the densities are 58 and 167, the centroid taken is 36.1125319434,-115.168108273 which is the orange point. All the other restaurants that are rated are represented using blue point. Hence one of either the green or the orange (depending on the density of the cluster) is taken as the reference location of the user. Finally, we take in the input of the location and compute the distance of the input location with all the users. We then select the ones which have the least distance and give them as the result.

9.3 Rating key attributes of a restaurant

In Figure 4, we performed LDA with 10 topics and assigned each topic to a certain attribute. For example, topics 4,7 and 9 correspond to Food attribute. Topics 3 and 6 correspond to Service attribute. Similarly, in Figure 5, we performed LDA with 5 topics.

In Figure 6, stars attribute represent the average ratings of the restaurant obtained from the dataset, whereas ratings for attributes like Food, Service, Value For Money and Ambience are averaged for each restaurant with the values we got from the sentiment analyzer.

10 CONCLUSION

We have developed an application to determine the local connoisseur given a user location i.e., we are providing a user with an option to find the local connoisseur so that users will know whom to approach to get the best recommendations about the restaurants in that locality. Future work for this project could be to to improve the model's performance by using Neural Network and other Machine Learning model.

This model is a good application for checking local connoisseur in a given locality.

For the second problem statement, where we are trying to evaluate the attributes of a restaurant, in the given time, we were able to develop a working model which performs the required task. Given more time, we can work more on the topic modelling and extract topics which will be more aligned to the attributes needed. Also, instead of defining the attributes, we can also mine the attributes from data itself. If we can generate a labeled dataset for this problem, we can put more efforts on improving the sentiment analysis and implement a model specific to the restaurant corpus.

Finally, we believe finding the rating of key attributes of a restaurant from reviews can be a very good application in real life which can help the user to get a better understanding of what is good and bad about a specific restaurant.

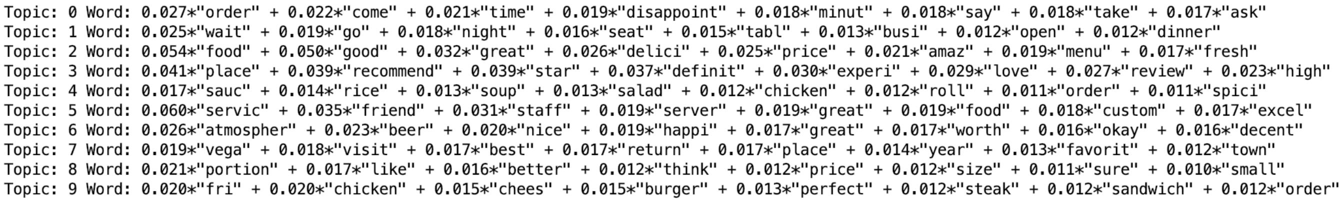
REFERENCES

[1] Vineeth Abraham. [n. d.]. Foodie Favorites. https://github.com/vabraham/foodie_favorites.

[2] Jennifer Lu Angela Gong. 2015. Picking Out Good Dishes from Yelp. https://nlp.stanford.edu/courses/cs224n/2015/reports/4.pdf.

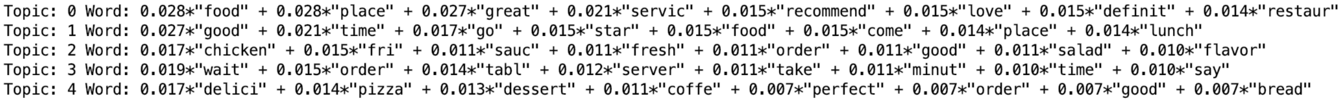
[3] Zhiyuan Cheng, James Caverlee, Himanshu Barthwal, and Vandana Bachani. 2014. Who is the Barbecue King of Texas?: A Geo-spatial Approach to Finding Local Experts on Twitter. In *Proceedings of the 37th International ACM SIGIR Conference on Research &#38; Development in Information Retrieval (SIGIR '14)*. ACM, New York, NY, USA, 335–344. https://doi.org/10.1145/2600428.2609580

[4] Jason P. C. Chiu and Eric Nichols. 2015. Named Entity Recognition with Bidirectional LSTM-CNNs. *CoRR* abs/1511.08308 (2015). arXiv:1511.08308 http://arxiv.org/abs/1511.08308



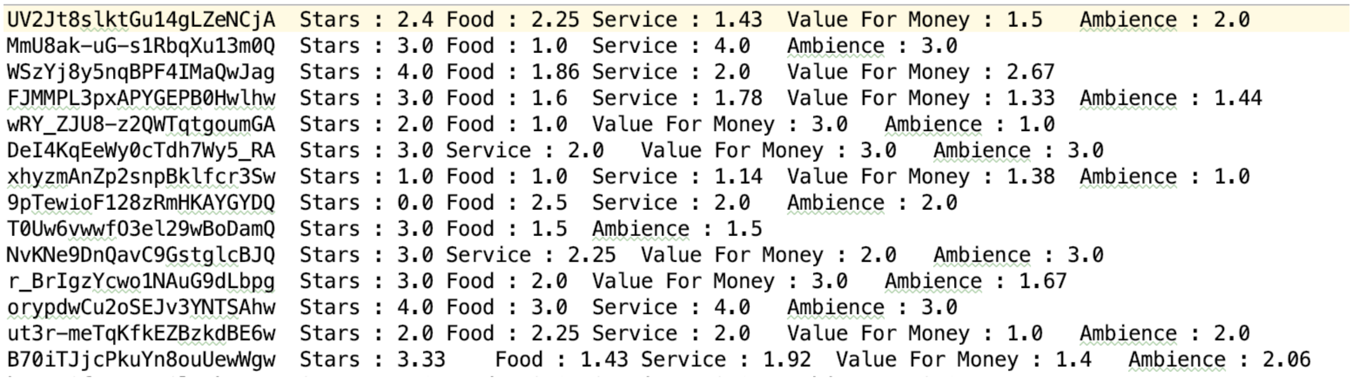
```
Topic: 0 Word: 0.027*"order" + 0.022*"come" + 0.021*"time" + 0.019*"disappoint" + 0.018*"minut" + 0.018*"say" + 0.018*"take" + 0.017*"ask"
Topic: 1 Word: 0.025*"wait" + 0.019*"go" + 0.018*"night" + 0.016*"seat" + 0.015*"tabl" + 0.013*"busi" + 0.012*"open" + 0.012*"dinner"
Topic: 2 Word: 0.054*"food" + 0.050*"good" + 0.032*"great" + 0.026*"delici" + 0.025*"price" + 0.021*"amaz" + 0.019*"menu" + 0.017*"fresh"
Topic: 3 Word: 0.041*"place" + 0.039*"recommend" + 0.039*"star" + 0.037*"definit" + 0.030*"experi" + 0.029*"love" + 0.027*"review" + 0.023*"high"
Topic: 4 Word: 0.017*"sauc" + 0.014*"rice" + 0.013*"soup" + 0.013*"salad" + 0.012*"chicken" + 0.012*"roll" + 0.011*"order" + 0.011*"spici"
Topic: 5 Word: 0.060*"servic" + 0.035*"friend" + 0.031*"staff" + 0.019*"server" + 0.019*"great" + 0.019*"food" + 0.018*"custom" + 0.017*"excel"
Topic: 6 Word: 0.026*"atmospher" + 0.023*"beer" + 0.020*"nice" + 0.019*"happi" + 0.017*"great" + 0.017*"worth" + 0.016*"okay" + 0.016*"decent"
Topic: 7 Word: 0.019*"vega" + 0.018*"visit" + 0.017*"best" + 0.017*"return" + 0.017*"place" + 0.014*"year" + 0.013*"favorit" + 0.012*"town"
Topic: 8 Word: 0.021*"portion" + 0.017*"like" + 0.016*"better" + 0.012*"think" + 0.012*"price" + 0.012*"size" + 0.011*"sure" + 0.010*"small"
Topic: 9 Word: 0.020*"fri" + 0.020*"chicken" + 0.015*"chees" + 0.015*"burger" + 0.013*"perfect" + 0.012*"steak" + 0.012*"sandwich" + 0.012*"order"
```

Figure 4: LDA using TF-IDF model for number of topics as 10.



```
Topic: 0 Word: 0.028*"food" + 0.028*"place" + 0.027*"great" + 0.021*"servic" + 0.015*"recommend" + 0.015*"love" + 0.015*"definit" + 0.014*"restaur"
Topic: 1 Word: 0.027*"good" + 0.021*"time" + 0.017*"go" + 0.015*"star" + 0.015*"food" + 0.015*"come" + 0.014*"place" + 0.014*"lunch"
Topic: 2 Word: 0.017*"chicken" + 0.015*"fri" + 0.011*"sauc" + 0.011*"fresh" + 0.011*"order" + 0.011*"good" + 0.011*"salad" + 0.010*"flavor"
Topic: 3 Word: 0.019*"wait" + 0.015*"order" + 0.014*"tabl" + 0.012*"server" + 0.011*"take" + 0.011*"minut" + 0.010*"time" + 0.010*"say"
Topic: 4 Word: 0.017*"delici" + 0.014*"pizza" + 0.013*"dessert" + 0.011*"coffe" + 0.007*"perfect" + 0.007*"order" + 0.007*"good" + 0.007*"bread"
```

Figure 5: LDA using TF-IDF model for number of topics as 5.



```
UV2Jt8slktGu14gLZeNCjA  Stars : 2.4 Food : 2.25 Service : 1.43  Value For Money : 1.5   Ambience : 2.0
MmU8ak-uG-s1RbqXu13m0Q  Stars : 3.0 Food : 1.0  Service : 4.0   Ambience : 3.0
WSzYj8y5nqBPF4IMaQwJag  Stars : 4.0 Food : 1.86 Service : 2.0   Value For Money : 2.67
FJMMPL3pxAPYGEPB0Hwlhw  Stars : 3.0 Food : 1.6  Service : 1.78  Value For Money : 1.33  Ambience : 1.44
wRY_ZJU8-z2QWTqtgoumGA  Stars : 2.0 Food : 1.0  Value For Money : 3.0   Ambience : 1.0
DeI4KqEeWy0cTdh7Wy5_RA  Stars : 3.0 Service : 2.0   Value For Money : 3.0   Ambience : 3.0
xhyzmAnZp2snpBklfcr3Sw  Stars : 1.0 Food : 1.0  Service : 1.14  Value For Money : 1.38  Ambience : 1.0
9pTewioF128zRmHKAYGYDQ  Stars : 0.0 Food : 2.5  Service : 2.0   Ambience : 2.0
T0Uw6vwwfO3el29wBoDamQ  Stars : 3.0 Food : 1.5  Ambience : 1.5
NvKNe9DnQavC9GstglcBJQ  Stars : 3.0 Service : 2.25  Value For Money : 2.0   Ambience : 3.0
r_BrIgzYcwo1NAuG9dLbpg  Stars : 3.0 Food : 2.0  Value For Money : 3.0   Ambience : 1.67
orypdwCu2oSEJv3YNTSAhw  Stars : 4.0 Food : 3.0  Service : 4.0   Ambience : 3.0
ut3r-meTqKfkEZBzkdBE6w  Stars : 2.0 Food : 2.25 Service : 2.0   Value For Money : 1.0   Ambience : 2.0
B70iTJjcPkuYn8ouUewWgw  Stars : 3.33    Food : 1.43 Service : 1.92  Value For Money : 1.4   Ambience : 2.06
```

Figure 6: Attribute ratings and actual ratings (Stars) for restaurants.

[5] Susan Li. 2018. LDA news headlines. https://github.com/susanli2016/NLP-with-Python/blob/master/LDA_news_headlines.ipynb.

[6] Wen Li, Carsten Eickhoff, and Arjen de Vries. 2014. Geo-Spatial Domain Expertise in Microblogs. https://www.researchgate.net/publication/260926996_Geo-Spatial_Domain_Expertise_in_Microblogs.

[7] Wikipedia. [n. d.]. Latent$_{D}irichlet_{a}llocation$. .