REFER: REstaurant Feedback from Existing Reviews

Fibin Francis Assissi
Computer Science
Northeastern University
francisassiassi.f@husky.neu.edu

Sankar Gireesan Nair
Computer Science
Northeastern University
gireesannair.s@husky.neu.edu

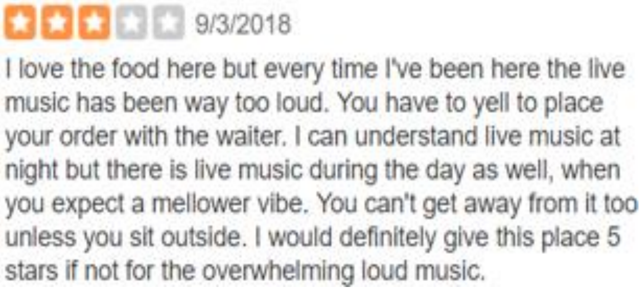
ABSTRACT

Online customer reviews play an important role in influencing consumer decisions. Restaurant customers are more likely to seek external information sources when they have not experienced it themselves. In this paper, we use data mining methodologies to extract features from user reviews and assign individual ratings to each of these features. Whereas traditional topic modeling approaches extract topics like *'food'* and *'service'*, our proposed model learns topics like *'good food'*, *'bad food'*, *'good service'* and *'bad service'*. In this paper, we propose a model in which term distribution of topics are also dependent on positive and negative lexicons in the review. Once these topics are extracted, we assign a rating for each topic which is higher or lower than overall rating depending on the positive or negative polarity of the topic respectively. The aggregated feature-wise ratings for a restaurant can provide meaningful insights to the restaurant owners.

1 INTRODUCTION

Yelp is a leading rating and review site for restaurants. For 80% customers, it is an integral factor in deciding where to eat. This has been a growing phenomenon over the past several years and studies have shown that reputation of a business on Yelp can dramatically affect its performance [1]. When people search about restaurants, the star rating usually decides whether a user will click to learn more about the restaurant or not. Also, research has shown that an extra half-star increase in rating causes a 19% increase in the likelihood of that restaurant getting fully occupied during peak hours. Because of these economic effects, it is critical for restaurant owners to understand what can be done to garner higher ratings on Yelp.

The rating of a restaurant on Yelp is the mean of all ratings given to the restaurant by users. However, is it enough to just evaluate the restaurant by a single star rating? Let us see the example of a review (Figure 1). In this case, the customer loves the food being served in the restaurant but the loud music being played affects the ambience of the place. The rating given by the customer is 3. But what the customer tries to say is that if it was for the food alone, the rating would have been 5 but because of the loud music overall rating got lowered to 3. We understand this only after reading the review.



★★★☆☆ 9/3/2018

I love the food here but every time I've been here the live music has been way too loud. You have to yell to place your order with the waiter. I can understand live music at night but there is live music during the day as well, when you expect a mellower vibe. You can't get away from it too unless you sit outside. I would definitely give this place 5 stars if not for the overwhelming loud music.

Figure 1: Sample rating and review from Yelp Website

If restaurant owners become successful in finding out opinion about different aspects of the restaurant, they would be able to raise their profits by improving on features they are lacking in. Also, the customers would be pleased because they get better experiences at restaurants. But it is not feasible for a restaurant owner to go through each review and find out what needs to be improved.

Our work focuses specifically on topic modeling methodologies to extract relevant features. But compared to traditional approaches, our model is based on the assumption that star ratings are an approximate function of positive and negative lexicons within the review. This approximation results in more fine-grained topic extraction such as *'good food'* and *'bad food'*, which provide better models of latent subtopics than those resulting from traditional approaches like LSA or LDA. By breaking down these reviews into relevant topics, we are then able to predict a restaurant's star rating per hidden topic. Ultimately these ratings per hidden topic allow us to pinpoint the reasons for a restaurant's Yelp rating.

2 RELATED WORK

In their work [2], J. Huang, S. Rogers, and E. Joo use LDA, an unsupervised learning algorithm that detects latent topics in a corpus and describes each topic with the probability of words occurring in the topic. It assumes that each review is made up of these topics and clusters the words into these K topics. After simple implementation, they presented the breakdown of hidden topics overall reviews, predicted stars per hidden topics discovered and extended their findings to that of temporal information regarding restaurants peak hours. They found several interesting insights, for example, they found that overall, the average rating of each hidden topic rating is lower if the overall rating of the restaurant is lower.
Methods like Latent Semantic Analysis are also used to describe the topics as mentioned in a paper by S. Deerwester [3]. The results for the LSA model are worse than that of LDA as can be seen in the Experiments section.

There are some other approaches that are very closely related to the problem being addressed. One of the approaches is based on sentiment analysis of the reviews. It just classifies the reviews as positive, negative or neutral without regarding the subtopics. Another approach uses traditional LDA to discover the hidden subtopics then assigns the same overall rating to each of the subtopics. One of the limitations with this method is that regardless of the polarity every subtopic gets an equal rating. The proposed model combines both these approaches - it extracts the subtopics from the review and recalculate the rating for each subtopic based on its polarity.

3 BACKGROUND INFORMATION

3.1 Topic Modeling - LSA and LDA

There has been a lot of work in the field of learning latent topics in a collection of documents. The first well known and popular topic modeling algorithm was Latent Semantic Analysis (popularly known as Latent Semantic Indexing) (LSI). LSA is an information retrieval technique which involves breaking the document by term matrix into topics and topic assignments using Singular Value Decomposition.

LDA is a common method of unsupervised learning to discover hidden topics. It assumes that there are latent variables that reflect the thematic structure of the documents (Blei, Ng & Jordan). It treats the probability distribution of each document over topics as a K-parameter hidden random variable rather than a large set of individual parameters (K is the number of hidden topics). With LDA, we can extract human-interpretable topics from a document corpus, where each topic is characterized by the words they are most strongly associated with. Furthermore, given a new document, we can obtain a vector representing its topic mixture, e.g. 5% topic 1, 70% topic 2, 10% topic 3, etc. These vectors are often very useful.

3.2 Term Frequency-Inverse Document Frequency (TF-IDF)

TF-IDF is a popular concept in information retrieval which gives the importance of a word in the document. Suppose we consider only the term frequency to represent the terms. In that case, the terms with the highest frequency would have the highest weight. But this can include stop words as well. Inverse document frequency, on the other hand, penalizes those words that appear in almost all the documents. It is based on the idea that rarer the word, more is the chance that it contains more information. IDF is calculated as the logarithm of the ratio of the total number of documents to the number of documents containing the term. TF-IDF is the TF multiplied by IDF makes TF-IDF value and it is used to represent the document as a vector which can be used in data mining and search engines.

4 PROPOSED APPROACH

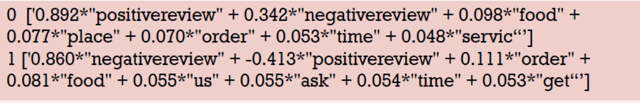
As mentioned in the previous sections, the problem statement can be broken down to two major parts: extracting the subtopic from the review and assigning an individual rating for each of the subtopics based on the polarity. Five different models were used - LSA with Positive/Negative processing, LDA TF-IDF, LDA TF-IDF with Positive/Negative processing, LDA with Positive/Negative processing and Two LDA models trained with positive and negative reviews separately.

The Yelp reviews from the dataset have undergone a set of preprocessing steps which include removal of stop words, stemming and tokenizing. After this, we perform an additional step in some models which we call as positive/negative processing. For this step, we extracted a total of 6000 domain-specific sentiment lexicons which has an equal proportion of positive and negative polarity lexicons. We then iterate through all the reviews and append a codeword after each sentiment lexicon. The codeword '*positivereview'* is appended next to a positive lexicon and *'negativereview'* is appended next to negative lexicon.

For e.g. The review "Food is good but service is bad" will be changed to "Food is *positivereview* good but service is *negativereview* bad". To choose a reasonable number of topics, we trained each of the models with different topic numbers.

4.1 Latent Semantic Analysis

LSA is a baseline model we used where the given set of input reviews are converted to TF-IDF matrix, where we have rows representing the terms in the corpus and the columns representing the review id. It then performs a singular value decomposition on this matrix. Table 1 shows the first 2 topics extracted by LSA model. It was implemented using Python Gensim package.



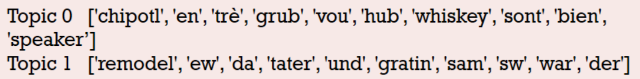
```
0  ['0.892*"positivereview" + 0.342*"negativereview" + 0.098*"food" +
0.077*"place" + 0.070*"order" + 0.053*"time" + 0.048*"servic“']
1 ['0.860*"negativereview" + -0.413*"positivereview" + 0.111*"order" +
0.081*"food" + 0.055*"us" + 0.055*"ask" + 0.054*"time" + 0.053*"get“']
```

Figure 2: Topics extracted by LSA

As observed from the results shown above, the same topic has a mix of words like *positivereview*, *negativereview*, food, service, etc. The topic has equal importance for contrasting words like *positivereview* and *negativereview*, which makes it difficult to interpret.

4.2 LDA TF-IDF

LDA is one of the most popular topic models. For this step, the additional step after preprocessing i.e. the appending of codewords after the positive/negative lexicon is not performed. The preprocessed reviews are converted to TF-IDF matrix and LDA is applied to extract the topics. The results obtained are as shown.



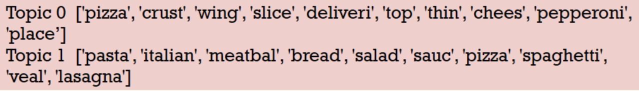
```
Topic 0   ['chipotl', 'en', 'trè', 'grub', 'vou', 'hub', 'whiskey', 'sont', 'bien',
'speaker']
Topic 1   ['remodel', 'ew', 'da', 'tater', 'und', 'gratin', 'sam', 'sw', 'war', 'der']
```

Figure 3: Topics extracted by LDA TF-IDF

As you can see, the common words like 'food', 'service', etc. are missing. This is because since they appear in almost all the reviews, their IDF weight is low resulting in very low TF-IDF value.

4.3 LDA TF-IDF with Positive/Negative Processing

In this model, we do the additional step of appending codeword after preprocessing. The input reviews are then vectorized using TF-IDF vectorizer and LDA is applied on top of it. We get the following results:



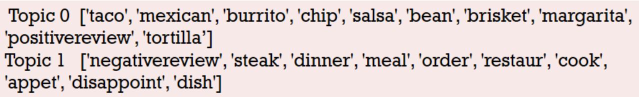
```
Topic 0  ['pizza', 'crust', 'wing', 'slice', 'deliveri', 'top', 'thin', 'chees', 'pepperoni',
'place’]
Topic 1  ['pasta', 'italian', 'meatbal', 'bread', 'salad', 'sauc', 'pizza', 'spaghetti',
'veal', 'lasagna']
```

Figure 4: Topics extracted by LDA TF-IDF with Positive/Negative processing

This model also does not include some common words in the restaurant reviews because of the low TF-IDF weights.

4.4 LDA with Positive/Negative Processing

Since TF-IDF was giving bad results, we switched to normal LDA which vectorizes the reviews based on frequency. The results are much better in this case as can be seen. Feature specific topics were identified from the generated LDA topics.



```
Topic 0  ['taco', 'mexican', 'burrito', 'chip', 'salsa', 'bean', 'brisket', 'margarita',
'positivereview', 'tortilla']
Topic 1  ['negativereview', 'steak', 'dinner', 'meal', 'order', 'restaur', 'cook',
'appet', 'disappoint', 'dish']
```

Figure 5: Topics extracted by LDA with Positive/Negative processing

Even though the model was generating meaningful topics, when we use the model to predict on a new review, topics generated represent both ‘*positivereview*’ and ‘*negativereview*’ for the same hidden feature, which was contradicting.

The limitation of the above LDA model to predict on a new review was solved by using a hybrid model which consider positive and negative topics separately.

4.5 Two LDA models trained with positive and negative reviews separately

In this model, two independent LDA models are used. Yelp reviews with a 5-star rating are considered as positive reviews and are used to train a positive LDA model which is used to identify positive topics. Each 5-star review is preprocessed and appended with the codeword *‘positivereview’* for all the positive lexicons present in the review. Yelp reviews with a 1-star rating are considered as negative reviews and are used to train a negative LDA model which is used to identify negative topics. Each 1-star review is preprocessed and appended with the codeword *‘negativereview’* for all the negative lexicons present in the review.

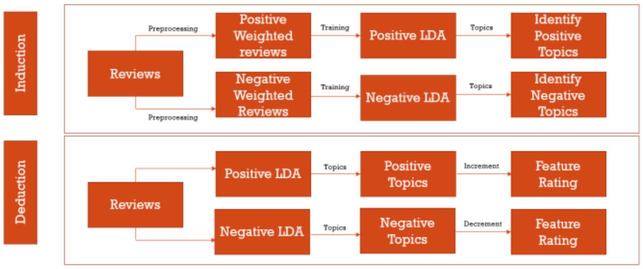


Figure 6: Model Architecture

The topics generated by these two trained LDA models are later categorized into different features of the restaurant. Topics which are a mixture of different features or which does not fall in any selected features are ignored.
All the topics from positive model is considered as positive review topic.

Feature Identified	Words in an Identified Positive Topic
Food	*['positivereview', 'restaur', 'menu', 'experi', 'food', 'dine', 'dish', 'chef', 'well', 'tast’]*
Service	*['positivereview', 'friendli', 'recommend', 'staff', 'food', 'place', 'great', 'super', 'highli', 'nice']*

Table 1: Output of Positive LDA Model

Similarly, topics from negative model is considered as negative review topic.

Feature Identified	Words in an Identified Negative Topic
Food	*['negativereview', 'food', 'place', 'tast', 'bad', 'like', 'ever', 'worst', ' bean', 'eat']*
Service	*['negativereview', 'us', 'ask', 'order', 'server', 'came', 'waitress', 'tabl', 'rude', 'back']*

Table 2: Output of Negative LDA Model

A new review is preprocessed and supplied to both models separately. If the generated topics from the review using a positive model is above the proposed threshold and falls in any of the already identified positive topics, the corresponding feature is assigned a rating higher than the overall rating for that review. If the generated topics from the review using a negative model is above the proposed threshold and falls in any of the already identified negative topics, the corresponding feature is assigned a rating which is lower than the overall rating.

4.5.1 *Workflow of the model with an example*

From our previous example, “Food is tasty, service is bad” be a review with rating “3”.

Step1: After preprocessing, the review becomes “Food *positivereview* good service *negativereview* bad”.

Step2: Output of step1 is fed to the positive LDA model. This will give a topic distribution which has higher value for the feature ‘*food’*. This implies that review has a positive sentiment towards feature ‘*food’* and hence the rating for feature *‘food’* is calculated as ‘3 + 1’ for this review.

Step3: Output of step1 is fed to the negative LDA model. This will give a topic distribution which has higher value for the feature ‘*service*. This implies that review has a negative sentiment towards feature ‘*service’* and hence the rating for feature *‘service’* is calculated as ‘3 - 1’ for this review.

5 EXPERIMENTAL SETUP

5.1 Datasets

The analysis presented in this paper uses Yelp dataset that was part of the Yelp Open Dataset challenge. It includes several subsections in JSON format - business, review, user, and rating.

The review object includes rating, review text, business id, and user id. From all the businesses, only restaurants with more than 100 reviews were filtered out. There are about 600,000 reviews from 180,000 restaurants.

Name	Size	Attributes Used
Review	600,000	Rating, Review text, Business ID
Business	180,000	Restaurant Rating, Business ID

Table 3: Specifications of dataset

In addition to this, a set of 6000 positive and negative sentiment lexicons were used which were extracted from the wordnet [5] for the positive/negative codeword appending.

Name	Size
Positive Lexicons	3000
Negative Lexicons	3000

Table 4: Size of Lexicons Used

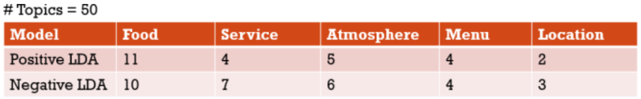
5.2 Preprocessing

The dataset was cleaned and only the useful attributes from review and business JSON objects were extracted. After this, a set of preprocessing steps were applied which include stop word removal, stemming and tokenizing. Along with this, punctuations and the numerical values were removed. The reviews are then subjected to additional processing of positive/negative codeword appending, depending on the model. After this optional step, word tokens are mapped to unique IDs to make the processing easier. This dictionary is saved so that reverse mapping can be done once we get the results. In addition to this, reviews are converted to Bag of Word representation using the Doc2Bow method provided by Gensim. Since we have a huge amount of data, the preprocessing and model generation were performed in AWS EC2 instance (m4.large machine), which made the work easier.

5.3 Result

5.3.1 *Number of Topics*

To choose a reasonable number of topics, the model was trained with different configuration and the topics generated were classified to dominant features.

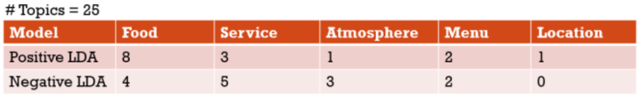


Topics = 50

Model	Food	Service	Atmosphere	Menu	Location
Positive LDA	11	4	5	4	2
Negative LDA	10	7	6	4	3

Figure 7: Topic distribution for 50 topics

When the model was trained with 50 subtopics, the subtopics which fall into the selected features are shown in figure 7. The subtopics were identified into 5 dominant features. Other topics were ignored.



Topics = 25

Model	Food	Service	Atmosphere	Menu	Location
Positive LDA	8	3	1	2	1
Negative LDA	4	5	3	2	0

Figure 8: Topic distribution for 25 topics

When the model was trained for 25 subtopics, the subtopics which fall into the selected features are shown in figure 8. When the number of topics was reduced from 50, the subtopics were becoming harder to interpret. Model with 50 subtopics was selected because the generated topics could easily be mapped to selected features.

5.3.2 *Predicting Feature-wise Rating*

100 Random restaurants with more than 100 reviews were selected. Each review was given to the models to generate topics which it belongs to. If the generated topics have a weight above the given threshold of 20% and it belongs to any one of the identified features, the respective feature rating was calculated. When a feature is identified from the positive model, it was assigned a rating which is equal to the original rating of the review +1 limited to [1,5]. When a feature is identified from the negative model, it was given a rating which is equal to the original rating of the review - 1 limited to [1,5]. The feature-wise rating for a particular restaurant is calculated by taking the average of ratings assigned to each feature for that restaurant.

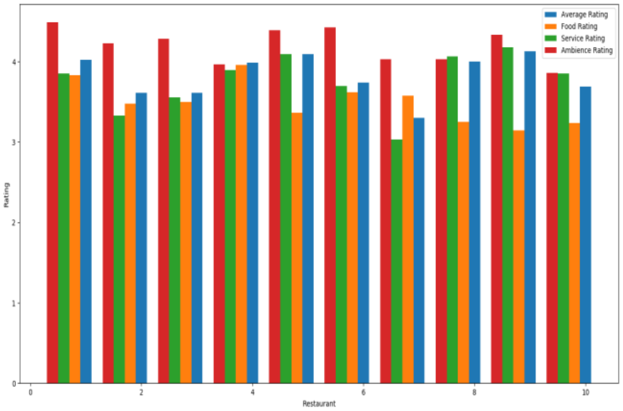


Figure 9: Comparison of average rating with feature-wise rating

Using this model, ratings for different features like *'Food', 'Service', 'Ambience', 'Menu', 'Location'* can be calculated for any given restaurant from its reviews. As shown in the figure 9, a feature-wise rating can be obtained and analyzed for improving the average rating of the restaurant. Food rating of restaurant 10 is below the average rating. In order to improve the average rating, the restaurant need to improve the quality of *'food'*. Similar analysis can be drawn for any restaurant.

5.3.3 *Result Analysis*

Graphs were plotted for average rating vs feature-wise rating. 10 random restaurants were selected for plotting the graph.

Average rating is the average of ratings given for reviews of a restaurant by users. Feature-wise rating is the rating of that feature for the restaurant calculated using the model.

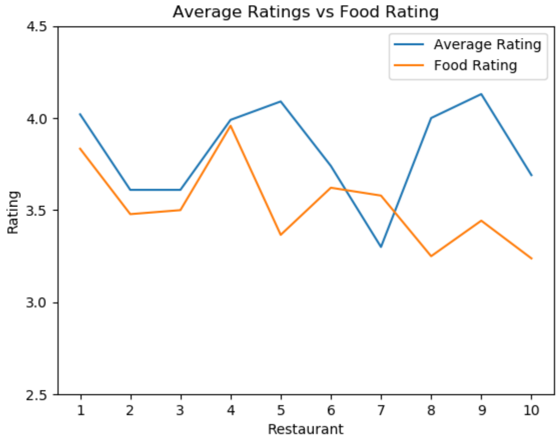


Figure 10: Comparison of overall rating with food rating

Figure 10 shows Average Rating Vs Food rating for different restaurants. The absolute difference between average rating and food rating falls within 0.5 star and they follow the same trend.

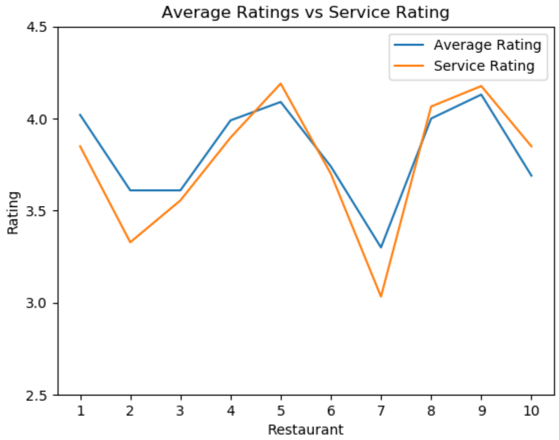


Figure 11: Comparison of overall rating with service rating

Figure 11 shows Average Rating Vs Service rating for different restaurants. Here also, the absolute difference between average rating and service rating falls within 0.5 star and they follow the same trend.

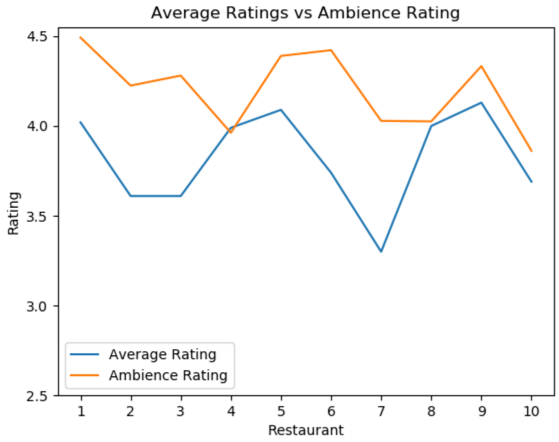


Figure 12: Comparison of overall rating with ambience rating

Figure 12 shows Average Rating Vs Ambience rating for different restaurants. Here, the ambience rating was higher than the average rating for most of the restaurants and they were not showing the same trend.

From the above analysis, it can be inferred that the average rating of a restaurant can be decomposed to feature-wise ratings which will give a good insight into the restaurant. It can be also seen that the average rating is always close to food-rating and service-rating. The other feature rating like '*ambience'* does not align with the average rating. From this, it can be concluded that *'Food'* and '*Service'* are the two dominant features that influence the average rating of a restaurant and improving them can increase the average rating significantly.

6 CONCLUSION

Compared to the traditional topic modeling approaches, the proposed model extracts the topics that can be easily interpreted as associated with positive or negative sentiment. This is helpful in calculating the feature-wise ratings**.** Overall, with the average feature-wise ratings for each restaurant, restaurant owners get better insights about each aspect of the restaurant. In addition to this, there are several other insights that can be drawn about the overall trend. One significant finding is that food and service are the dominant factors in deciding the overall rating of a restaurant.
Since this is an unsupervised learning model, we had to manually decide on the performance of each of the model based on the interpretability and correctness of the topic. For evaluating the model, a set of reviews of a restaurant were taken and the feature-wise ratings were calculated manually. The same set of reviews, when inputted to the model, gave results that align with the manually calculated results.

7 FUTURE WORK

The additional processing step implemented for the final model used the sentiment lexicons extracted from Wordnet. But some of the restaurant jargons were missing in this set. To overcome this, we are interested in developing a semi-supervised approach to collect domain specific lexicons. Initially, we will collect the lexicons from a set of 1-star and 5-star reviews. This will be used as a seed set to collect more lexicons using Turney algorithm.

Additionally, instead of using a general model for all the restaurants, we can develop restaurant specific models. For e.g. The reviews for a Pizza place will be completely different from that of a Chinese restaurant. Our approach will perform better if we have restaurant-specific models compared to a generic model for all the restaurants.

8 REFERENCES

[1] https://www.hotel-online.com/press_releases/release/online-customer-reviews-their-impact-on-restaurants
[2] J. Huang, S. Rogers, and E. Joo, "Improving restaurants by extracting subtopics
from yelp reviews," iConference 2014 (Social Media Expo), 2014.
[3] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman,
"Indexing by latent semantic analysis," Journal of the American society for
information science, vol. 41, no. 6, p. 391, 1990.
[4] Huang, J., Rogers, S., & Joo, E. Improving restaurants by extracting subtopics
from yelp reviews. iConference 2014 (Social Media Expo). (2014)
[5] https://wordnet.princeton.edu/