Semi-Supervised Semantic Review Aggregation

Saket Sharma, Swaroop Gadiyaram and Venkatesh Elango

Abstract— **Rating prediction is an important part of today's e-commerce applications. A lot of research has been done in predicting the ratings (or stars) of individual reviews for commodities and businesses using the review text. However, in this project, we propose a method to predict the rating of a business (restaurant) itself. We use the text form reviews using several models and combinations of models based on Latent Dirichlet Allocation (LDA) for topic modeling, Bag-Of-Words (BOW) and term frequencyinverse document frequency (*tf-idf*) and predict restaurant stars using Gradient Boosting for Regression (GBR). Comparison of our models is done in terms of MSE. Our evaluations suggest that a text based model is potentially useful for predicting the ratings for new businesses with fewer reviews. We use the dataset from *Yelp Dataset Challenge 2017* for our predictive task. Finally, we show the potential for using topic modeling for sub-category and other attribute prediction for restaurants. Sub-categories and attributes, such as coffee shops, desserts, breakfast, dietary options etc., are defined in the Yelp dataset.**

Keywords: Yelp, Business Rating Prediction, Topic Modeling, LDA, GBR, *tf-idf*, BOW.

I. INTRODUCTION

There has been great enthusiasm in the domain of rating prediction and recommender system design in the last few years in academia and industry alike, especially since the *Netflix Challenge* and the *Yelp Dataset Challenge*. A lot of research has been done in predicting the star rating given by a user to an item or a business given the review text.

A lot of information can become obscured in large reviews, so most review hosting services like *Yelp* also provide users with an option to just give a star rating along with the review text. This method, however, is prone to subjective bias. Two users may review a business positively in their text, however, they may still have different ratings, in which case, their positive review opinion can get obscured. A method, that predicts business ratings using the text of the review alone can get a rating, more representative of the actual experience, and eliminate user bias. In this project, we propose such a model that predicts business ratings based on the raw text of the review. This approach, also helps predict a more accurate rating for new businesses with fewer reviews.

II. RELATED WORK

A lot of research has been conducted in the domains of web information extraction systems [5], phrase sentiment orientation analysis for reviews [6] and opinion modeling [7]. Our work combines opinion modeling with the use of Latent Dirichlet Allocation (LDA) for topic modeling[1].

Latent topic modeling is widely used as an unsupervised model for clustering and classification in both natural language processing, computer vision and other machine learning areas and is used to discover hidden topics. Besides its use for generating hidden topics for a corpus, LDA has been used to predict ratings for reviews by several authors. Huang et al use a traditional LDA to discover hidden topics [8], and then predict stars for these hidden topics. Mingming Fan and Maryam Khademi attempted to predict star ratings of a business using the text of the review alone[11]. However, they did not use LDA for topic modeling, tested their model on a much smaller dataset, and did not try out Gradient boosting regression for prediction and hence, reported much higher MSE.

Yatani et al. [9] and Huang et al. [10] designed different interfaces for Yelp that show top frequent adjectives used to describe a business. The papers do not go into details into the effectiveness of using sentiment scores over raw review text.

In our work, we explore the effectiveness of using hidden topics uncovered by using LDA on several combinations of Parts of speech (POS) of sentences in reviews like nouns, adjectives, adverbs etc. Figures 11 and 12 show the words for two topics inferred from fitting an LDA model. As can be seen, the words from the topics are strongly correlated to the reviewer's experience at the restaurant. So, these topics can be used to predict ratings for the restaurant itself.

III. Background

A. Latent Dirichlet Allocation (LDA)

Latent Dirichlet Allocation (LDA) [1] is a Bayesian generative model for text. It is used as a topic model to discover the underlying topics that are covered by a text document. LDA assumes that a corpus of text documents cover a collection of K topics. Each topic is defined as a multinomial distribution over a word dictionary with $|V|$ words drawn from a Dirichlet $\beta_k \sim Dirichlet(\eta)$.

Each document from this corpus is treated as a bag of words of a certain size, and is assumed to be generated by first picking a topic multinomial distribution for the document $\theta_d \sim$ Dirichlet(α). Then each word is assigned a topic via the distribution θ_d, and then from that topic k, a word is sampled from the distribution β_k. θ_d for each document can be thought of as a percentage breakdown of the topics covered by the document.

The topic distribution of a corpus from the LDA model can be found in numerous ways. With the LDA model [1], Blei et al. also present an Expectation Maximization algorithm that converges to the most likely parameters (word distributions per topic and topic distributions per word). For this project , we are using the LDA implementation from *gensim*[4].

B. Term frequency - Inverse document frequency (tf-idf)

In a large text corpus, some words will be very frequent (e.g. the, a, is in English) hence carrying very little meaningful information about the actual contents of the document. If we were to feed the count data directly to a classifier those very frequent terms would shadow the frequencies of rarer yet more interesting terms. In order to re-weight the count features into values suitable for usage by a classifier it is very common to use the *tfidf* transform. For this project we use the *scikit-learn* implementation of *tf-idf* [12].

IV. Exploratory Data Analysis

For this project, we use the latest Yelp dataset that is provided for the 2017 *Yelp Dataset Challenge*. The dataset includes information about local businesses, reviews and users in cities from USA, UK, Germany and Canada. The dataset contains reviews for businesses from 1143 categories, including restaurants, theaters, dermatologists, zoo, health insurance offices etc. Figure 1 shows the top 20 most frequent businesses in the dataset. As can be seen, restaurants are the most reviewed business in the *Yelp dataset* with 33907 restaurants. Since we are processing the review text assuming it to be in English, we choose to use restaurants only from the US. This reduces our set to 20880 restaurants. The dataset is split into 16704 restaurants for training and 4176 restaurants for testing. Since every restaurant has multiple reviews, the actual number of reviews in training and test dataset are 1.55 million and 330,000 respectively. The only hyper-parameter in our task is the number of topics chosen to represent the reviews for LDA (Section III). However, due to lack of computational resources, we cannot cross validate this parameter on validation sets of substantial size. We have chosen our number of topics to be 250 because of computational bottlenecks and also because we want to visualize the learned topics. We are using a gradient boosting regression model for our prediction which is an ensemble method that is robust to overfitting.

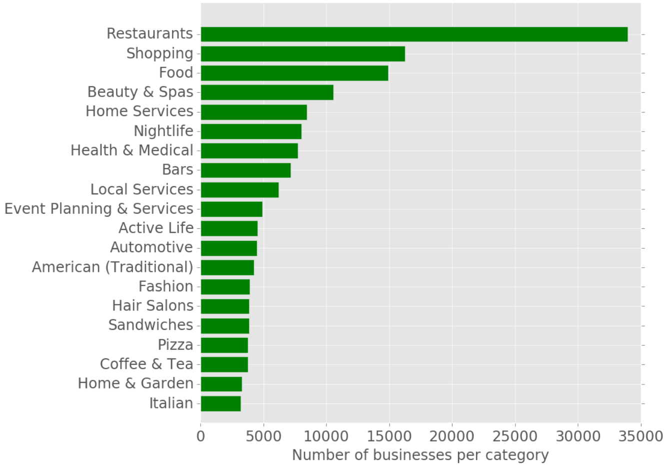

Fig. 1. Top 20 most represented business categories

Figure 2 shows the relation between restaurant star rating and the number of reviews. We expected the star rating to be higher with the increasing number of reviews and it generally seems to be the case except for very high rated restaurants. Possible reason for this could be that those restaurants are either new or expensive.

Figure 3 shows the correlation between the restaurant star rating and average of user ratings for a restaurant. We expected Yelp to be doing the average of user ratings to predict restaurant star rating. But, we can see that Yelp is not just doing this simple averaging to predict restaurant star rating. One reason could be because of the rounding that Yelp does. As mentioned in the Section I, for restaurants with fewer reviews, this way of prediction might encode individual user biases.

Figure 4 shows a histogram of restaurant and review

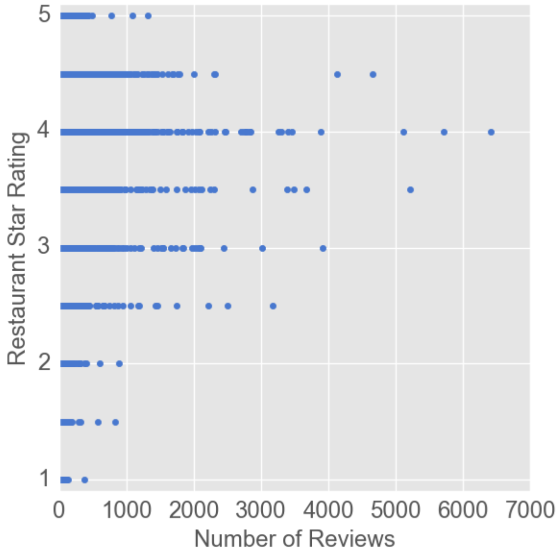

Fig. 2. Restaurant Star Ratings vs Number of Reviews

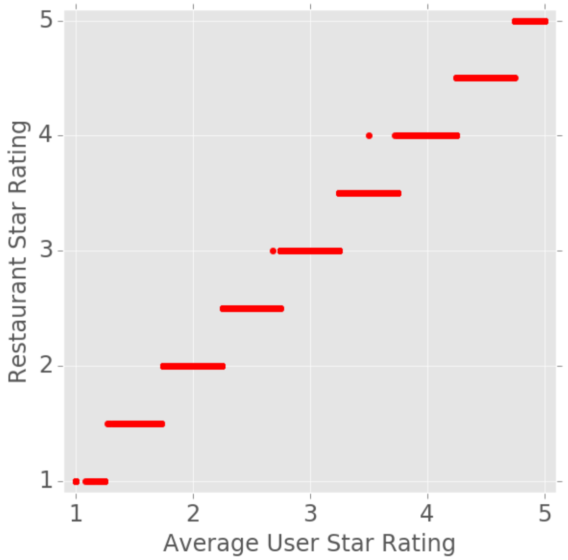

Fig. 3. Restaurant star ratings vs User Average Star Rating

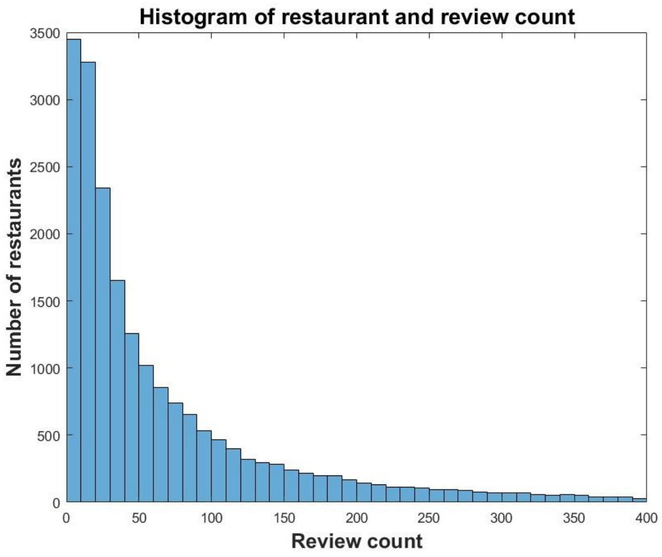

Fig. 4. Histogram of restaurant and review counts

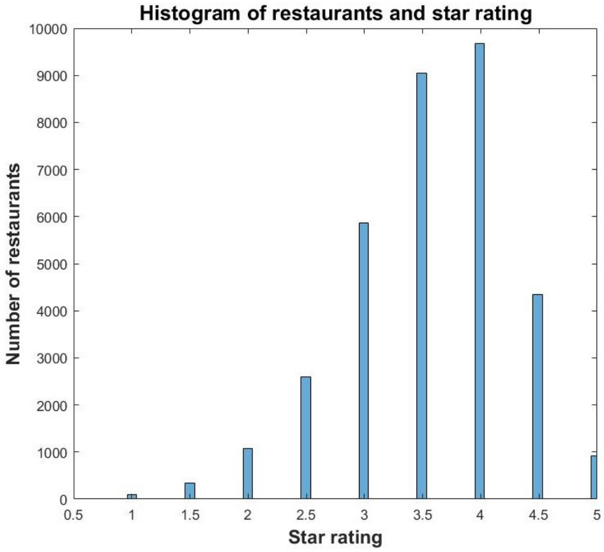

Fig. 5. Histogram of restaurants and star rating counts

counts. We observe that most of the restaurants have very few reviews. This motivates our model that gives good predictions for ratings of restaurants with limited number of reviews.

Figure 5 shows the histogram of restaurants and star ratings rounded to half stars (as in the Yelp dataset). As can be seen, over 55% of the restaurants have ratings of 3.5 or 4.

V. Predictive task

As discussed in Section I., we are trying to predict the restaurant rating from the text of the user reviews alone. The details of baselines, choice of features and model are discussed in Section VI.

VI. Model

We choose the following models for the baselines for our predictive task and for the actual prediction:

A. Baselines

1) Average star rating: For this baseline, we decide to use the average of the star ratings of all the restaurants that are available in our data set and in the United States.

2) City-wise average star rating: Herein, we calculate the average of the star ratings for all the restaurants for every city in the training set and then for a restaurant in the test set predict its star rating as the average star rating from the city it is in.

3) Zip-wise average star rating: For this baseline, we calculate the average of the star ratings for all the restaurants for every zip-code in the training set and then for a restaurant in the test set predict its star rating as the average star rating from the zip-code it is in. If we have no other restaurants in a zip-code we use the city-wide average.

B. Latent Dirichlet Allocation

We do some preprocessing of the review text before fitting LDA model. Part-of-speech tagging (POS tagging) is the process of reading a text and marking each word as corresponding to a particular part of speech, based on both its definition and its context, such as noun, adjective, verb, adverb etc.

For extracting topics from reviews, we remove stopwords, POS tag each review (using NLTK) and then retain the words belonging to a subset of all the tags. In this paper we consider the following 2 subsets:

- nouns
- nouns, adjectives, and adverbs.

Since we want to extract topics such as type of cuisine and service quality from reviews, we decided to use nouns. We then added adjectives and adverbs to the list of tags retained because we postulated that adjectives and adverbs would describe a topic in a better way than using only nouns–for example, we would want *horrible service* and *great experience* to be in separate topics.

After POS tagging all the reviews and retaining the words belonging to the subset of tags under consideration, we build an LDA model and extract 250 topics from the reviews. Next,we aggregate the results per restaurant as we want to get a feature vector to predict the star rating for every restaurant. We propose the following two ways of doing this aggregation: LDA topics based aggregation, and LDA topic words aggregation.

1) LDA topics based aggregation: The aggregation algorithm is as follows

- For *each* restaurant in the training set:
  - For each review corresponding to that restaurant, using the previously trained LDA model obtain the probability distribution over the topics.
  - Compute the geometric mean of the predicted probability distribution for reviews.
  - Store this geometric mean as the feature vector for that restaurant.

We assume that given the restaurant, topic distribution of reviews are independent, thus taking the geometric mean gives the probability distribution over topics for a given restaurant normalized by the number of reviews. We do this because we want our algorithm to predict the star ratings even when there are very few reviews.

2) LDA topic words aggregation: The aggregation algorithm is as follows

- For *each* restaurant in the training set:
  - Generate vocabulary using top k words for each topic using the previously trained LDA model.
  - For each review corresponding to that restaurant predict the probability distribution over the topics. For each topic we get the top k words and weigh each word with topic probabilities.
  - Sum the word probabilities across the reviews for a given restaurant and save that as a feature vector. The feature vector will be of the size of vocabulary.

We propose this model to have a continuum between using purely the topics in the reviews and purely using the word counts in reviews. This model obtains the word probabilities in a Bayesian framework which is in contrast to the frequentist inference from *tf-idf*.

C. Bag of words (BOW)

We describe our bag of words based feature extraction algorithm below.

- For *each* restaurant in the training set:
  - Concatenate all reviews corresponding to that restaurant into a single vector after removing punctuations and stopwords.
- Build a bag of words model over the training set which counts and retains the most frequently occurring 2000 words across the entire training data.

D. Term frequency – inverse document frequency (tf-idf)

We describe our *tf-idf* based feature extraction algorithm below.

- For *each* restaurant in the training set:

  - Concatenate all reviews corresponding to that restaurant into a single vector after removing punctuations and stopwords.
- Build a bag of words model over the training set which counts and retains the most frequently occurring 2000 words across the entire training data.
- Normalize the bag of words frequency counts by using term frequency and inverse document frequency.

It is crucial to note that when bag of words or tf-idf for feature extraction we provide information about the number of reviews. Thus, these approaches might not work well when a newly opened restaurant which has very few reviews.

E. *Regression Model*

After extracting the features from the above mentioned techniques, we use Gradient Boosting Regressor [2], [3] as our regression model to predict the star ratings. Besides trying out these models separately, we also concatenate the features from various models and use that as the feature vector for the regression model.

VII. Results and Observations

We list the results in Table 1, for the models that we proposed in Section VI. We see that using the average star rating as the prediction gives an MSE around 0.52. This baseline is quite strong and the reason can be inferred from Figure 5: over 55% of the restaurants have a star rating of 3.5 or 4.

It can be seen that our proposed models outperform the baseline by a significant margin. Amongst our proposed models, *tf-idf*+LDA topics (Nouns) gave an MSE of 0.122 edging out the *tf-idf* model. Due to limitation in our compute power we were not able to evaluate the *tf-idf*+LDA topics (Nouns, Adjectives, Adverbs) model. Since using LDA topics model (with 250 topics) using nouns, adjectives and adverbs give an MSE of 0.22, we believe that increasing the number of topics would bring the MSE to be on par with our *tf-idf*+LDA topics model. Due to the tremendous increase in computational complexity, we weren't able to train LDA words model on the entire dataset. So, we used a subsample of 100,000 reviews to train LDA words model (using only nouns) and got MSE of 0.342. We believe that the LDA words model, if trained on the full dataset will be better than LDA topics model and also expect that using the adjectives as well as adverbs would reduce the MSE.

In the following word clouds we qualitatively show the probability of each word given the topic, i.e, the larger the size of a word the more probable it is for that topic. In Figure 6 we show the probability distribution over words for a topic – which we identify as *Pizza* based on the words shown. We also show Figures 7, 8, 9, 10, 11, 12 we show the probability distribution over words for various topic and manually name the topics. We postulate that these topics can also be used for predicting:

- categories, such as the type of restaurant (Figures 6, 7, 8)
- attributes, such as the dietary options, good for meal (Figures 9, 10)
- star ratings (Figures 11, 12).

TABLE I
Performance of various models

Model	**MSE**
Average Star rating	0.526
City-wise average star rating	0.527
Zip-wise average star rating	0.520
LDA Topics (Nouns, Adjectives, Adverbs)	0.226
LDA Topics (Nouns)	0.304
LDA Words (Nouns)	0.342*
BOW	0.125
tf-idf	0.123
tf-idf + LDA Topics (Nouns)	**0.122**
BOW+LDA Topics (Nouns)	0.131

**Trained on 100,000 reviews only*

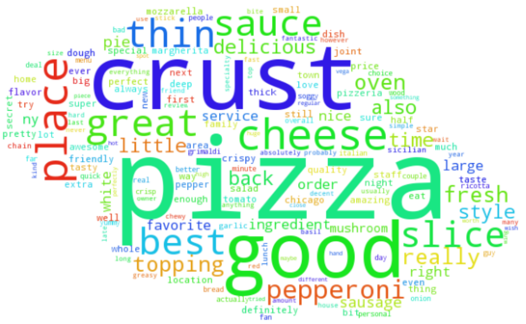

Fig. 6. Word Cloud for topic "Pizza"

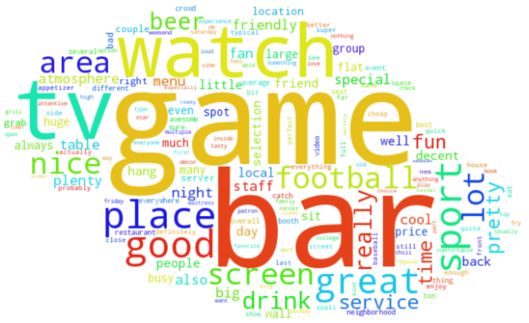

Fig. 7. Word Cloud for topic “Sports Bar”

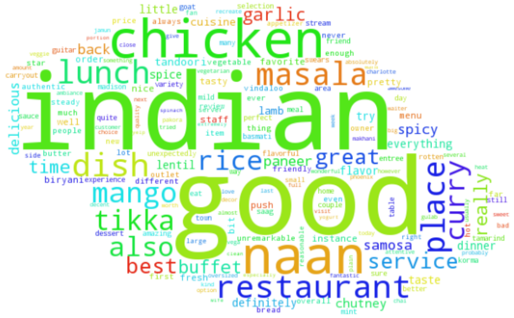

Fig. 8. Word Cloud for topic “Indian food”

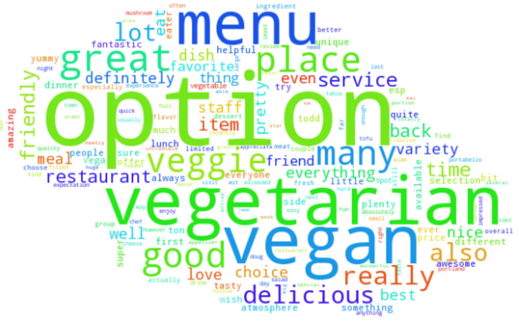

Fig. 9. Word Cloud for topic “Dietary Options”

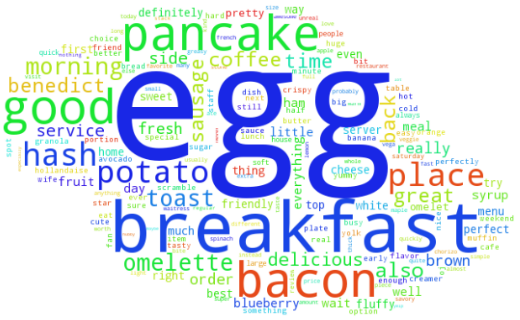

Fig. 10. Word Cloud for topic “Breakfast”

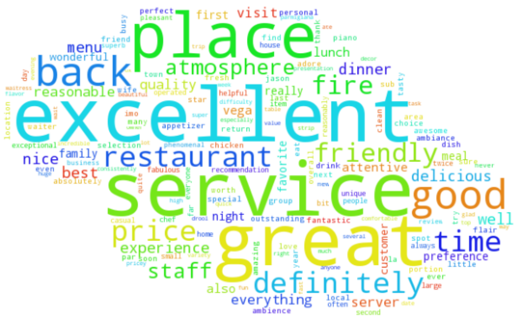

Fig. 11. Word Cloud for topic “Good”

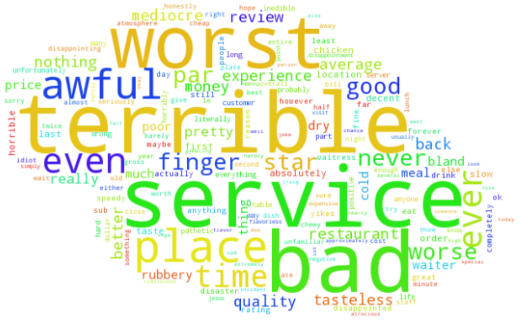

Fig. 12. Word Cloud for topic “Bad”

VIII. CONCLUSIONS

In this paper we studied the effectiveness of using review text for predicting star rating of a restaurant. We tested several different models for using the text such as LDA with topics, LDA with words, bag of words, *tf-idf* etc. We also experimented with using different combinations of POS for generating features. From our experiments, we observed that combining *tf-idf with LDA topics (using only nouns) based aggregation* gave the minimum MSE for our predictive task. The model performs really well even for restaurants that do not have many reviews. Our primary contribution is the proposed LDA topics based aggregation model, which can be learned in an unsupervised manner and the learned topics could be used for other supervised learning tasks such as predicting other attributes of a business, including, but not limited to subcategories such as type of restaurant. Given better computational resources, we would test the rating predications using LDA topic words based aggregation, LDA topics based on bigrams model as features.

REFERENCES

[1] David M. Blei, Andrew Y. Ng, Michael I. Jordan, *"Latent Dirichlet Allocation"*, Journal of Machine Learning Research 3 (2003) 993-1022

[2] J. Friedman, *Greedy Function Approximation: A Gradient Boosting Machine*, The Annals of Statistics, Vol. 29, No. 5, 2001.

[3] J. Friedman, *Stochastic Gradient Boosting*, 1999.

[4] Rehurek, Radim and Petr Sojka, *"Software Framework for Topic Modelling with Large Corpora"*, Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks, 2010.

[5] C.-H. Chang, M. Kayed, M. R. Girgis, and K. F. Shaalan, *A Survey of Web Information Extraction Systems,* IEEE Trans Knowl Data Eng, vol. 18, no. 10, pp. 14111428, Oct. 2006.

[6] B. Pang, L. Lee, and S. Vaithyanathan, *Thumbs up?: sentiment classification using machine learning techniques,* in Proceedings of the ACL-02 conference on Empirical methods in natural language processing - Volume 10, Stroudsburg, PA, USA, 2002, pp. 7986.

[7] B. Pang and L. Lee, *Opinion Mining and Sentiment Analysis,* Found Trends Inf Retr, vol. 2, no. 12, pp. 1135, Jan. 2008.

[8] J. Huang, S. Rogers, and E. Joo. *Improving restaurants by extracting subtopics from yelp reviews* 2014.

[9] K. Yatani, M. Novati, A. Trusty, and K. N. Truong, *Review spotlight: a user interface for summarizing user-generated reviews using adjective-noun word pairs,* in Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, New York, NY, USA, 2011, pp. 15411550.

[10] J. Huang, O. Etzioni, L. Zettlemoyer, K. Clark, and C. Lee, *RevMiner: an extractive interface for navigating reviews on a smartphone,* in Proceedings of the 25th annual ACM symposium on User interface software and technology, New York, NY, USA, 2012, pp. 312

[11] Mingming Fan and Maryam Khademi, *"Predicting a Business Star in Yelp from Its Reviews Text Alone"*

[12] Pedregosa et al. *Scikit-learn: Machine Learning in Python*, JMLR 12, pp. 2825-2830, 2011