CS 585 Project Final Report:
Performing Sentiment Analysis on 3-Star Restaurant Ratings

Nikhil Garg, Shivangi Singh

December 2017

1 Abstract

Industrialization has changed our lives in numerous ways. It has lead to the growth of the food industry. On average, 72% of Americans eat lunch at a food joint on a daily basis. With the increased number of competitors, it is essential for a business to have good reviews written about it if its owners want to expand their business. So we have tailored our idea around a business to help them increase their ratings by targeting the neutral reviewer. It is easier to convince a neutral person to like a commodity than to convince a person who is already against the product. How do we know which customer is neutral? Simple - we just assume that all neutral reviewers leave a rating of 3 stars.

2 Objective

Our project had two main objectives. The first objective for us was to perform an experiment where we try to label 3-star reviews as either positive or negative. The second objective for us was to perform an experiment where we try to determine the most positive and most negative aspects of a restaurant where the aspects are extracted from 3-star reviews.

3 Introduction

Sentiment Analysis is the task of classifying opinions in text as either positive or negative (or in some cases, neutral). Aspect-based sentiment analysis is the task of extracting certain features from the text and performing sentiment analysis on those features. An example of a feature is "food" in a restaurant review; the author's opinion of food can be classified as either positive or negative (or in some cases, neutral). In our project, we intend to classify aspects of the restaurant reviews authored by Yelp reviewers as either positive or negative, without neutral as an option. For this we used various techniques/packages such as the TextBlob Naive Bayes and Sentiment Analyser, NLTK pos tagging and our own Customized

Naive Bayes and Turney's Algorithm. We then predicted the negative/positive aspects of a business based on the polarity scores of the Nouns. Example we found that milk had a negative score.

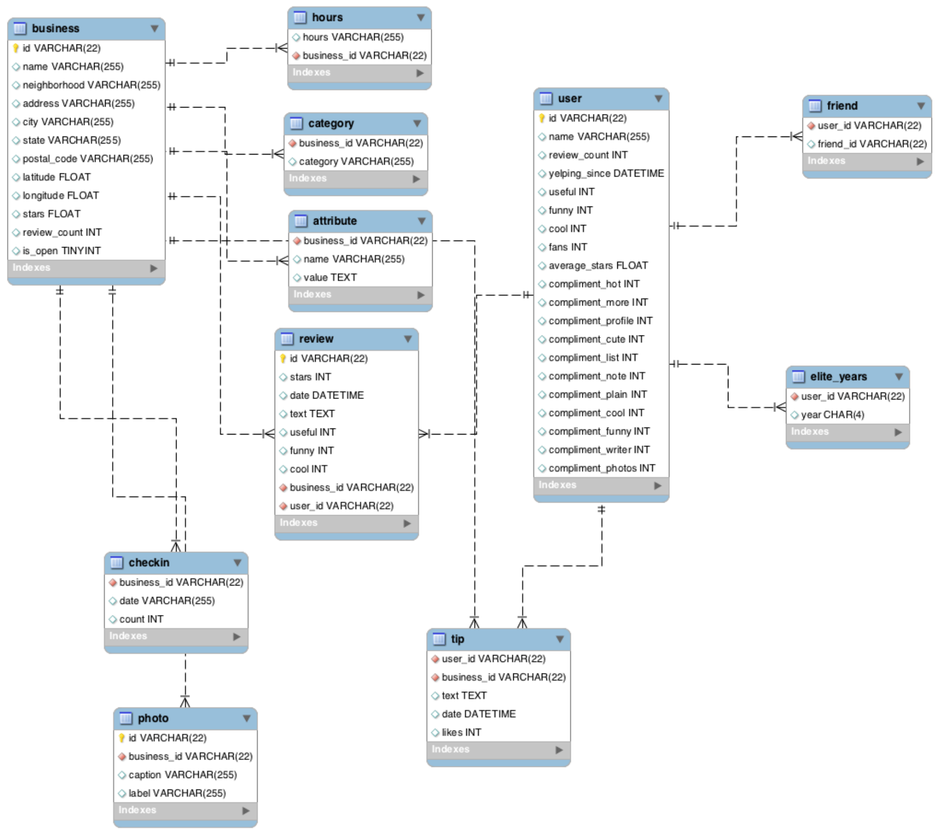
The proceedings of 2002 Conference on Empirical Methods in Natural Language Processing(EMNLP) Pang et al. describe the successful use of traditional machine-learning techniques, such as Naive-Bayes, for the sentiment classification of entire movie reviews. [2] We can use the same technique on our Yelp dataset to tag the 3-star reviews and predict it to be positive or negative. But this bag-of-words technique is not that accurate for sentences that don't have words like "bad" and "good". An example is "Is this place even worth a hype?". When a human reads this question, he or she can predict that the review is a negative review. The bag-of-words technique cannot do the same. So we are going to try to improve our prediction by using Turney's [3] unsupervised algorithm that uses bi-grams containing adjectives or adverbs to make a prediction.

In the experiment that Turney performed, he used 410 reviews from Epinions. His algorithm attains an average accuracy of 74%. It appeared that movie reviews were difficult to classify, because the whole is not necessarily the sum of the parts; thus the accuracy on movie reviews is about 66%. On the other hand, for banks and automobiles, it seems that the whole is the sum of the parts, and the accuracy is 80% to 84%. Travel reviews are an intermediate case. But we think that this should work better on our dataset because people don't really write long reviews about food on Yelp.

4 Dataset

For our project we are using the Yelp Dataset which is easily available to us on Yelp as part of their Dataset Challenge. The data contained information about local businesses in 12 metropolitan areas across 4 countries

Figure 1: Schema of the Yelp Dataset



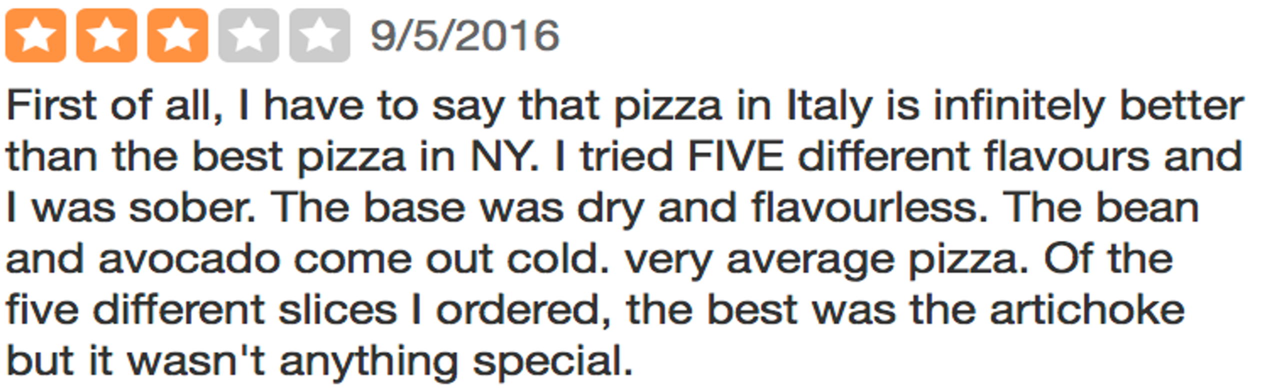
There were two uncompressed files that we needed for our experiments: review.json and business.json. Business.json had the list of all the businesses in the dataset, and review.json had the list of all the reviews for all the businesses in the dataset.

We downloaded these files and parsed each line as JSON objects and we parsed the JSON objects using Python. Then we queried the dataset to select for restaurant reviews and businesses only. As there are no special parameters in the Yelp Dataset for classifying a business as a restaurant or not, we filtered out the reviews for which the business categories included 'Food'.

Figure 2: The size of the Yelp Dataset

Size of Yelp Dataset		
Unit	**Compressed**	**Uncompressed**
Gigabytes	**2.28**	**5.79**
Files	**1** (.tar.gz)	**6** (.json)

An example of a review from the dataset is as follows:



5 Method

Naive Bayes

The Naive Bayes classifier is a generative classifier, which means that given an observation, the classifier returns the class mostly likely to have generated that observation.
Naive Bayes works under two assumptions:
1) Bag of Words Assumption: which assumes that the position of a word in a context doesn't matter.
2)Conditional Independence: Assumes that the feature probabilities $P(x_i|c_j)$ are independent given class c. $P(x_1, x_2, ...x_n|c) = P(x_1|c) * P(x_2|c) * ..... * P(x_n|c)$ The Naive Bayes Model is based on the maximum likelihood estimate which simply uses the frequencies in the data

$$P(c_j) = \frac{reviewcount(C=c_j)}{N_{review}}$$

$$P(w_i|c_j) = \frac{count(w_i,c_j)}{\sum count(w,c_j)}$$

over the vocabulary.

As it turns out, TextBlob has its own implementation of the Naive Bayes classifier. The TextBlob Naive Bayes was trained on a movie review set. We expected this classifier to work relatively poorly because the sentences framed for reviewing movies is really different from the ones used for reviewing food/restaurants. So we created our own customized Naive Bayes model and trained it on the restaurant reviews set. We did take into consideration that this method is biased against words not seen in the training corpus, as their probability is reduced to zero. To counteract that, we have used alpha smoothing i.e. giving every word an initial count of alpha. This is different from the Naive Bayes done by the TextBlob package as it ignores the words not seen in the training corpus.

In order to train the customized Naive Bayes model, we treated reviews with more than 3 stars as positive examples and reviews with less than 3 stars as negative examples. Basically, we are assuming that reviewers who rated a restaurant with more than 3 stars leave mostly positive reviews, while reviewers who rated a restaurant with less than 3 stars leave mostly negative reviews. We extracted 1000 positive reviews as our training set and 100 (separate) positive reviews as our test set. Similarly, we extracted 1000 and 100 negative reviews as our training and test sets, respectively. We can use the same collection for training and test sets because the labels are already there: they are the star ratings, and they indicate whether a review should be treated as positive or negative. Thus, we don't have to manually tag each review in the test set as positive or negative, and this saves us a lot of time.

Turney's (2002) Sentiment Orientation Model

Thirdly, we implemented Turney's (2002) Sentiment orientation model on our reviews. Turney's model analyzes full sentences rather than words. We used positive and negative seeds from an annotated customer review dataset provided by Hu and Lui(2004) (https://www.cs.uic.edu/ liub/FBS/sentiment-analysis.html). We then used Turney's algorithm to find the polarity associated with each word which was found by the count of their occurrence alongside a seed word. The polarity of the word is calculated thusly:

$$Polarity(word) = PPMI(word, positive_word) - PPMI(word, negative_word)$$

where PPMI is calculated as so:

$$PPMI(w, s) = max(\frac{logP(w,s)}{P(w)P(s)}, 0)$$

where w is the word and s is the seed. Where PPMI is the Positive Point-wise Mutual Information of a word. We have used max(0, p) as to account for words that are encountered for the first time.) We have used the method to narrow down the top 10 most negative and top 10 most positive words to predict the characteristics of the business, i.e. if food has a negative polarity we can tell the business that the food is bad and it is something that the business could work on. Similarly if music has a positive polarity we can predict for the business that music is a positive aspect of their business. As we are limiting the top most negative/positive words to be nouns it is likely that we will find names of dishes showing up that would be more meaningful for the business as then they can selectively improve upon the dishes.

Own Implementation

To further our aspect based prediction about the business. We took to our advantage that various aspects like music, dishes names, ambience would be tagged as Nouns or some variations of nouns by the NLTK pos-tagger. So, we made a dictionary for those nouns whose value is another dictionary of adjectives that occur along with the noun in a sentence, the dictionary of adjectives have a count associated with it. The count describes the number of occurrence of the adjective and the word together in a sentence.As we are parsing sentences by breaking on punctuation our method works fairly well in associating nouns with their adjectives. Although this method is limited to querying the most commonly used adjectives for a given Noun and doesn't consider negations and sentences split by a punctuation like a comma.

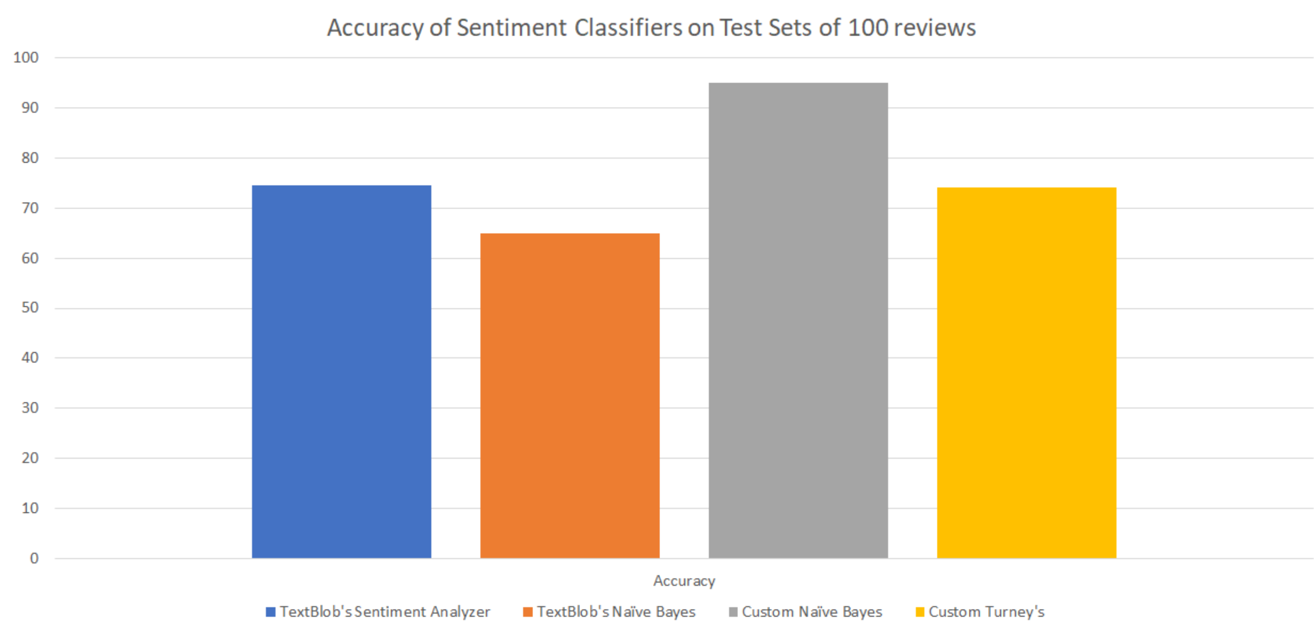
6 Result

6.1 Experiment 1

Figure 3: Accuracy for the TextBlob's Pattern Analyzer, TextBlob's Naive Bayes Analyzer, Customized Naive Bayes, Customized Turney's Method

Accuracy	
TextBlob's Pattern Analyzer	**74.5**
TextBlob's Naive Bayes Analyzer	**65.5**
Custom Naive Bayes Classifier	**95.4**
Custom Turney's Method	**70.0**

As can be seen, the customized Naive Bayes Classifier which was trained on restaurant reviews (similar model) worked much better than the baseline model trained on movie reviews. This is probably due to the fact that movie reviews and food/restaurant reviews have different characteristics and sense in how they are worded. It also worked much better than the Textblob's pattern analyzer, which only had an accuracy of 74.5 as compared to the customized Naive Bayes model's accuracy of 95.4. It is obvious that using a training set related to the item we are classifying works better for the classification of the item.



6.2 Experiment 2

Turney's (2002) Sentiment Orientation Model

Once again, we used Turney's Sentiment Orientation model in this experiment. The difference between how we use this model in this experiment versus how we used it in experiment 1 is that this time we will be using the model to determine which aspects of a restaurant are polarized in the reviews, and then using the polarity scores calculated using PPMI (as explained in the previous section), the model will classify each polarized aspect of the restaurant as either positive or negative based on the review.

For predicting the positive aspect for a restaurant we queried the top ten words with a positive and negative polarities respectively. The results were as follows:

Figure 4: List of top ten polarized words in positive and negative category

Polarized words	
Positive Polarity	**Negative Polarity**
teas	**thru**
youre	**orders**
pot	**mediocre**
Te	**store**
music	**driver**
ambience	**customer**
Cafe	**im**
truck	**Starbucks**
price	**bread**
everyone	**card**

Looking at the list of positive words we can see words (aspects) like tea, pots, music , ambience, price are legitimate words describing a businesses aspect like most three star reviews might have rated the business music,ambience, tea, pots collections to be good. So on seeing this the business can feel secure that they are doing well in those areas. Similarly, looking at the negative words list words like driver, card, bread, orders can indicate that the driver most likely for a delivery messed up, or the bread was bad, and orders were messed up. Looking at such works the businesses can look for areas where they can improve like in this case get better quality bread, or check up on the delivery driver.

Own Implementation

The various aspects of businesses like dishes names, music, decorations etc. would be tagged as nouns by the NLTK pos-tagger. So we used that to our advantage to query the most frequently used adjectives for a given noun. For this section we queried different nouns that a business might be interested in.

Figure 5: List of top five adjectives queried for a given Noun(aspect).

Noun	Adjectives
chicken	**good**
	soggy
	entree
	favorite
	decent
service	**good**
	great
	quick
	slow
	bad

As we can see that this works well in filtering out the adjectives describing the aspects. But this is limited to the number of reviews we parse, and the number of times the exact adjective is used to describe the noun and the tagger correctly identifies it, but it works fairly well.

7 Error Analysis

In order to analyze how well our results were, we had to use some sort of metric to determine how accurate our results were. We decided to compare our results in both experiments to our own interpretations of the sentiments of the 3-star reviews. Here is an excerpt from a 3-star review:

If you need an inexpensive place to stay for a night or two then you may consider this place but for a longer stay I'd recommend somewhere with better amenities. Pros: Great location- you're right by the train station, central location to get to old town and new town, and right by sight seeing his tours. Food, bars, and shopping all within walking distance. Location, location, location. Very clean and very good maid service Cons: Tiny rooms Uncomfortable bed Absolutely no amenities No phone in room No wardrobe Was given a lot of attitude about me and my husband sharing a room which was quite strange and we were charged 15 pounds more for double occupancy not sure why that matters I felt like it was a money grab. It was just handled in a kind of odd manner to me... If you book this hotel all you get is a bed, desk, and a bathroom. It isn't awful but know what you're getting into.

So from just reading this review one can notice that though this review is technically neutral as a whole, it tends to lean farther on the negative side of the spectrum (one way to tell is that there are a lot more phrases containing negative words than there are phrases with positive words).

The methods that implemented had differing outcomes when they ran on this review. Our baseline Naive Bayes model, Textblob's Naive Bayes analyzer, labeled this review as positive, while our own Naive Bayes model labeled this as negative. We guessed that the baseline model probably looked at the words "inexpensive", "Great", "clean", and "good" as the key words to determining that this review was positive. It is difficult to say exactly why it made a mistake as the analyzer was trained on a different dataset than the one used in our experiment.

When performing error analysis on the results of the second experiment, we found that the results were correct for the most part. The results indicated that words like "teas" and "music" had a positive polarity associated with them in the same reviews as words like "mediocre" and "bread", which had a negative polarity associated with them. When we manually checked the reviews to see if the words were labeled correctly by our methods, we found that our methods had been almost entirely accurate.

8 Future Work

For sentiment analysis, bag-of-words model is not a good method because location and interactions between words matter. For example, negative sentiment are often expressed with the negations of positive words, with modifier words such as "not" and "never".
A suitable model to counteract this problem of negation we can use is the recursive neural tensor network, which has shown commendable performance in sentiment analysis. We also plan on using the Convolutional neural network (CNN) from Kim Y (2014) [1], which only requires sentiment at the sentence level which could be extracted from the reviews by delimiting on punctuation.
We have to build a UI for taking in a business ID and querying that business reviews accordingly. As we are trying to market this idea for a specific business to predict the restaurants sentiments.

References

[1] Yoon Kim. "Convolutional Neural Networks for Sentence Classification." In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (2014), pp. 1746–1751. DOI: 10.3115/v1/d14-1181.

[2] et al Pang Bo. "Thumbs up?" In: *Proceedings of the ACL-02 conference on Empirical methods in natural language processing - EMNLP 02* (2002), pp. 79–86. DOI: 10.3115/1118693.1118704.

[3] Peter D Turney. "Thumbs up or thumbs down?" In: *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics - ACL 02* (2001), pp. 417–424. DOI: 10.3115/1073083.1073153.