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

Nikhil Garg, Shivangi Singh

November 2017

1 Scope and Objective

In this project, we will be performing two experiments. In the first experiment, we will perform basic Sentiment Analysis on restaurant reviews where the reviewers rated each restaurant with 3 out of 5 stars. We aim to classify "neutral" 3-star reviews definitively as either positive or negative. In the second part of our experiment, we will perform aspect-based sentiment analysis on 3-star reviews. In this experiment, we intend to extract aspects about the restaurant from each 3-star review such as "food" and "service" and then perform sentiment analysis on each feature to determine whether reviewers thought aspects of the restaurant such as "food" were either good or bad.

2 Background

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.

3 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 size of the Yelp Dataset is represented in this table:

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

We downloaded the dataset 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'.

4 Method: Experiment 1

We implemented two baseline classification models and two customized classification models for the first experiment of our project.

Textblob's Pattern Analyzer

For our first experiment, we decided to use the Python package TextBlob to implement our baseline models. TextBlob has a built-in pattern analyzer to analyze text for sentiment. It does so by looking at words from the famous word corpus WordNet. WordNet has thousands of words with a polarity ("positive" or "negative") attached to each word. The pattern analyzer checks the text to see how many positive words from the corpus in the text and then subtracts it by how many negative words there are from the corpus. If the result is negative, it classifies the text as negative, and if the result is positive, it classifies the text as positive.
We use TextBlob's Pattern Analyzer to analyze our reviews for sentiment. Basically, for each review in our positive training set, we determine if the review is positive or negative using the pattern analyzer. Ditto for the negative training set. Positive and negative training examples were chosen based on whether they had more than 3 stars or less than 3 stars, respectively. Then we calculate the amount of correct answers the patter analyzer gave over the total number of reviews in the training sets, and that is our accuracy score. Our training sets in this case had 1000 reviews each.

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.

5 Preliminary Results: Experiment 1

Here we have tabulated our preliminary results:

Preliminary Results: Accuracy		
TextBlob's Pattern Analyzer	**TextBlob's Naive Bayes Analyzer**	**Custom Naive Bayes Classifier**
74.5	**65.5**	**95.3846153846**

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.

We haven't finished implementing Turney's method yet so we haven't included them in the results table.

6 Method: Experiment 2

In the second experiment, we plan on using at least two different models (hopefully three) to calculate the sentiments of the different aspects of the restaurants.

Turney's (2002) Sentiment Orientation Model

Once again, we plan on using 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.

Recursive Neural Tensor Network (RNTN) Socher (2013)

We also plan on using a recursive neural tensor network model and comparing it to our customized Turney method in addition to a customized Convolutional Neural Network model.

7 Preliminary Results: Experiment 2

We haven't run the models for experiment 2 quite yet, so we unfortunately do not have results to show for the second experiment in this stage.

8 Remaining 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), 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.

Estimated Time of Remaining Work

Given our time constraints and meeting availability, we plan on finishing Turney's method by the end of Thanksgiving break. We plan on getting the CNN working by December 2, 2017 and will possibly finish the implementation of RNTN by December 7, 2017 which will give us a few days to make changes for our final presentation.

9 Bibliography

Kim Y. (2014) Convolutional Neural Networks for Sentence Classification, Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP) pp. 17461751, Doha, Qatar, October 25-29, 2014
Socher R. et al. (2013) Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank, Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP)
Peter D Turney. "Thumbs up or thumbs down?" In:Proceedings of the 40th Annual Meeting on Association for Com-putational Linguistics - ACL 02(2001), pp. 417–424.doi:10.3115/1073083.1073153.