**Sentiment Analysis: YELP Dataset Restaurant Review**

**Final Project - CS6320 Natural Language Processing**

Sneha Bangar

**Abstract**

Yelp is a website which publishes crowd sourced reviews about local businesses (Restaurants, Bars, Cafes, Department Stores, Home-Local Services, and Automotive etc.). Yelp online reviews are invaluable source of information for users to choose where to visit or what to eat among numerous available options. However, due to overwhelming number of reviews, it is almost impossible for users or restaurant owners to go through all reviews and find the information they are looking for. Thus, restaurants need a way to interpret their reviews. By providing a solution to relate review text to rating, we can help restaurants understand how they would be rated.

To provide an overview of a business, one solution is to give use business star rating of 1-5 star(s). However, this rating can be subjective and biased toward use’s personality. In this project, a restaurant’s rating is predicted based on user-generated review’s texts only. This helps to provide an overview of long review text but at the same time cancels out the subjectivity. The purpose of this project is to build an algorithm to perform sentiment analysis on Yelp Restaurant Review Dataset and identifying the tone of the user reviews for a restaurant as positive, negative or neutral. This also involves tagging restaurant as good/average/bad on the basis of the percentage of positive, negative or neutral reviews.

**I. INTRODUCTION**

Sentiment analysis has become an important research area for understanding people’s opinion on a particular matter by analyzing a large amount of information. Billions of people express their opinions about different services or products using popular review sites, social networking sites, or blogs. This active feedback of the people is valuable for companies to analyze their customer’s satisfaction and the monitoring of business competitors, also it is of significant importance for consumers who want to research a product or a service prior to making a purchase or visit.

Yelp consists of a large amount of business and user data such as number of user check-ins, user reviews, business star rating and business categories for a large number of states in United States and some other parts of world. Yelp user reviews express opinions and sentiments about businesses and service providers among a given rating, scaled from 1 to 5, which is used as a general metric review. The purpose of the yelp restaurant reviews is to inform the reader about the restaurant of the review, most likely to make some decision about whether to visit it or not based on how good/bad the reviews are. However, often it is the case that there are hundreds or thousands of reviews available for a single restaurant, making it too tedious for one person to read all reviews for that restaurant. This presents a problem for the reader of the reviews, who wants to gain as much relevant information as possible before making some decision about the restaurant. Also it might be the case that, many customers have described the restaurant with multiple positive words such as “perfection”, “must go”, “great treat”, “tasted great” etc. in their review but some customers have given star rating of 5 and others have given rating of 3. Thus, it became the motivation to predict a restaurant’s rating based on its reviews text to reduce the bias of the reviewers.

The project focuses to solve the reader’s problem by processing the information contained in all of the user generated reviews and identify the positive/negative/neutral tone of the review. This in turn will help to identify the rating of the restaurant. The solution in this report rates a restaurant as Good/Average/Bad from just the text of its reviews. It helps to learn the best mapping from a review word vector to a restaurant rating. This solution will provide advanced analytics to current business owners to grow and improve their businesses.

**II. PREVIOUS WORK**

Previous work regarding sentiment analysis classification using machine learning techniques in determining if the overall sentiment of a review is positive or negative used movie reviews as data. The authors use a unigram model and Naïve Bayes, maximum entropy classification, and support vector

machines to perform the sentiment classification and achieve 80% accuracy. They concluded that their results outperform the method based on human tagged features [2].

Previous work using sentiment analysis and Yelp dataset reviews focused on predicting star rating using the text alone. The authors experiment different machine learning algorithms such as Naïve Bayes, Perceptron, and Multiclass SVM on a sample of 100 000 user reviews from Yelp dataset. They use

Bing Liu Opinion Lexicon for feature selection and some preprocessing techniques such as removing stop words or stemming (i.e. reducing the words to their root form). The best result (precision and recall) is obtained using Naïve Bayes and feature selection with stop words removed and stemming [3].

In [4] in order to capture word sentiments the authors used a supervised learning algorithm based on similarities between words which takes into account the rating of previous reviews for capturing the representation of words vectors. On a dataset of about 20 000 unique reviews from Yelp dataset, the

accuracy reported is about 70%.

**III. DATA**

Yelp has publicly released a sample of their data (including over 2.7 million reviews) as part of their [Dataset Challenge](http://www.yelp.com/dataset_challenge/). This data can be used for the project as it is easy/quick to acquire. This solution uses the data sets provided by Yelp in the Yelp Dataset Challenge. It is available online at Yelp dataset, located at [1]

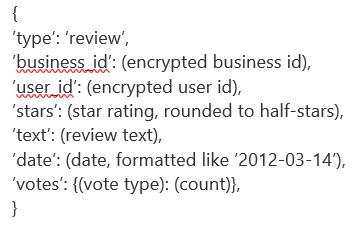
It includes the data

* 2.7M reviews and 649K tips by 687K users for 86K businesses
* 566K business attributes, e.g., hours, parking availability, ambience.
* Social network of 687K users for a total of 4.2M social edges.
* Aggregated check-ins over time for each of the 86K businesses
* 200,000 pictures from the included businesses

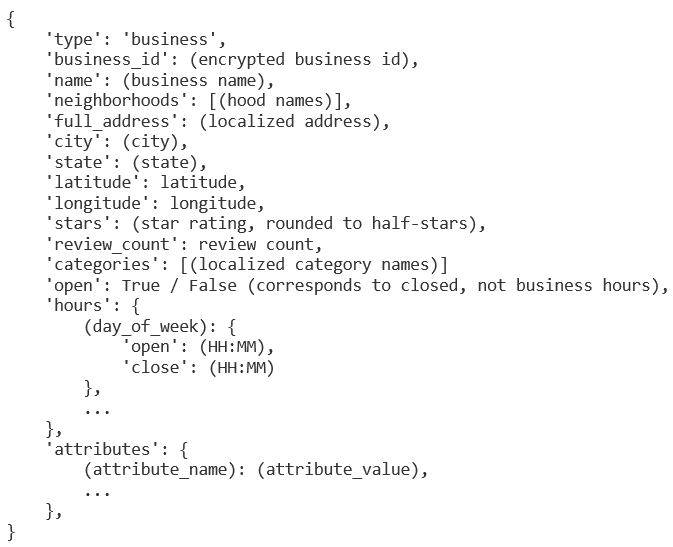
The data for the project was taken from following two files

* yelp\_academic\_dataset\_review.json
* yelp\_academic\_dataset\_business.json

All the data are in json format. For reviews, the data looks like



For business, the data looks like



**IV. PROPOSED METHOD**

**Defining Sentiment**

For the purpose of project, we define sentiment to be "a personal positive or negative feeling." Here are some examples:

|  |  |
| --- | --- |
| **Sentiment** | **Review** |
| Positive | The food here is very good. |
| Neutral | The ambience is okay and the food was usual. |
| Negative | I am never coming to this restaurant again. The food was tasteless. |

**High Level Steps**

The high-level sequence involved in processing is as follows:

1) Raw data collection from Yelp Dataset

2) Sentiment labeling

3) Transform into train/test sets for classifier

4) Bag of Words

5) Transform train/test sets for final classification by classifier

6) Adjust classifier and repeat until best model

**Transformation**

This project focuses only on business raw text reviews and their star ratings for sentiment classification. For this project, Python inbuilt JSON library was used to read the data. The data from business and review JSON files was extracted and stored in two Pandas DataFrames. The businesses were filtered based on their ‘category’ field and only businesses having category as “Restaurants” were fetched.

In order to use supervised learning and train a classifier, we normally require a hand labeled training data, but considering the large number of reviews, it would be very difficult to manually annotate the data to train a sentiment classifier for reviews. This project assumes that the star rating is an accurate measure for the sentiment opinion of the review. The star rating of a business review is an integer from 1 to 5. I have decided to consider all the reviews with ratings 1-2 as “negative”, all ratings 4 and above as “positive” and reviews with star rating 3 as “neutral” sentiments.

For this project, 15k reviews from the dataset have been used. After splitting the filtered Yelp challenge dataset in 80% for training and 20% for testing, preprocessing techniques in order to extract a set of features. This data was further used to train classifiers like Naïve Bayes and Maximum Entropy Model.

**Preprocessing**

*Tokenization*

Tokenization is the process of chopping up sentences into smaller pieces (words or tokens). The segmentation into tokens can be done with decision trees, which contains information to correctly solve the issues you might encounter.

For each review in the training set following operations were performed-

* Strip the newline “\n” at the end of each review.
* Place a space before and after each of the following characters:

.,() []:;”

(This prevents sentences like “I like this restaurant.It is classy” being interpreted as [“I”, “like”, “this”, “restaurant.It”, “is”, “classy”].)

* Tokenize the text by splitting it on the basis of spaces.
* Remove tokens which consist of only a space, empty string or punctuation marks.

*Word Normalization*

Word Normalization is the reduction of each word to its base/stem form. While doing this, following issues were considered:

1. Capital letters should be normalized to lowercase.
2. Handling apostrophe (‘); “Joey’s Book” should obviously be tokenized as “Joey” and “Book”, but I’m, we’re, they’re should be translated as I am, we are and they are.
3. Handling words like lower-case, U.S.A.

*Stop Word Removal*

In order to check accuracy of prediction with stop words Removal (common words in English, but with no sentiment information) from the text reviews Porter corpus of stop words from Natural Language Toolkit (NLTK) was used.

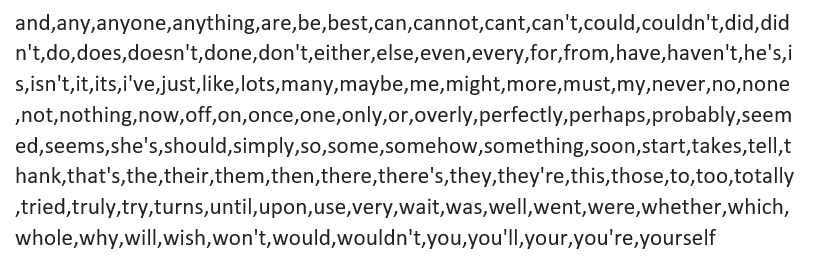
*Bag of Words*

This project uses Bag of words approach for feature selection. The bag-of-words model is one of the simplest language models used in NLP. It makes a unigram model of the text by keeping track of the number of occurrences of each word. This can later be used as a features for Text Classifiers. In this bag-of-words model you only take individual words into account and give each word a specific subjectivity score. If the total score is negative the text will be classified as negative and if its positive the text will be classified as positive. It is simple to make, but is less accurate because it does not take the word order or grammar into account. Steps Followed -

* To keep track of the number of occurrences of each word, tokenized the text and added each word to a single list. Then by using a Counter element kept track of the number of occurrences.
* Made a DataFrame containing the class probabilities of each word by adding each word to the DataFrame as we encounter it and dividing it by the total number of occurrences afterwards.
* These words in this constructed Sentiment Lexicon were used to give a value to the subjectivity of the reviews in the test set.

*N-Gram*

To include features consisting of two or three words, so that “very good” and “not good” will be two different features with different subjectivity scores, a small set of words (conjunctions, prepositions, interjections etc.) was defined, which changes the meaning of the words following it or the rest of the sentence. Whenever a n-gram word is encountered, the sentence is not split but it is split after the next word. In this way n-gram features consisting of the specified words and the words directly following them were constructed. Following words have been used for this purpose:



**Classifiers**

Two different classifiers were used. Naive Bayes classifier and maximum Entropy Model. A Naive Bayes classifier was built from scratch. Third-party library Natural Language Toolkit (NLTK) was used for Maximum Entropy.

Classifier Evaluation

A clean and unambiguous way to present the prediction results of a classifier is to use a use a confusion matrix (also called a contingency table). For this project classification, the table has 3 rows and 3 columns. Across the top is the observed class labels and down the side are the predicted class labels. Each cell contains the number of predictions made by the classifier that fall into that cell.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Predicted Condition** | | | |
| **True Condition** | **Total Population** | **positive** | **negative** | **neutral** |
| **positive** | True positive | False negative | False neutral |
| **negative** | False positive | True negative | False neutral |
| **neutral** | False positive | False negative | True neutral |

Table 1: Contingency table

For determining the accuracy of a single Classifier, we use accuracy which represents the percentage of test samples that are classified correctly from all test samples.

Accuracy (ACC) = Σ True positive + Σ True negative/Σ Total population

**Naïve Bayesian**

Naive Bayes is a simple model for classification. It is simple and works well on text

categorization. This project adopts multinomial Naive Bayes. It assumes each feature is

conditionally independent to other features given the class. That is, where C is a specific class and t is text we want to classify. P(C) and P(D) is the prior probabilities of class and text. And P (D | C) is the probability the text appears given this class. In this project the value of class C might be Positive, Negative or Neutral. Here D refers to all of the text in the entire training set. It is given by D = (d1, d2, .., dn), where di is the ith attribute (word) of document D. The probability of a class C is given by the posterior probability P(C|D) given a training document D. The goal is choosing value of C to maximize P(C | D)

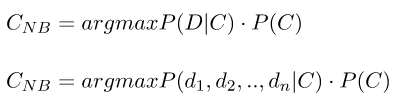
Using Bayes’ rule, this posterior probability can be rewritten as



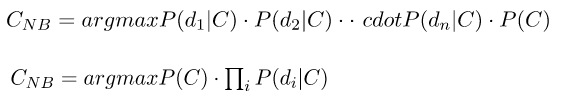
Since the marginal probability P(D) is equal for all classes, it can be disregarded and the equation becomes



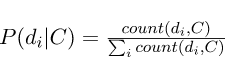
The document D belongs to the class C which maximizes this probability, so



Assuming conditional independence of the words di, this equation simplifies to



Here P(di | C) is the conditional probability that word i belongs to class C. For the purpose of text classification, this probability can simply be calculated by calculating the frequency of word i in class C relative to the total number of words in class C.



The prior-probabilities were estimated with a training set with documents that are already labeled with their classes. Using this training set, trained the model and obtained values for the prior probabilities. This trained model was used for classifying unlabeled documents.

Let’s consider a sample example, say we have counted the number of words in a set of labeled training documents. In this set each text document has been labeled as either Positive, Neutral or as Negative. The result will be as follows:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Word | Positive Class | Neutral Class | Negative Class | Total |
| restaurant | 10 | 90 | 10 | 100 |
| fantastic | 70 | 20 | 10 | 100 |
| lame | 10 | 20 | 70 | 100 |
| this | 50 | 500 | 50 | 600 |
| is | 100 | 600 | 100 | 800 |
| Total | 240 | 1230 | 240 | 1700 |

Table 2: Sentiment Lexicons

From this table we can already deduce each of the class probabilities:

P(Cpos) = 0.141, P(Cneu) = 0.723, P(Cneg) = 0.141.

If we look at the sentence “This restaurant is fantastic”, then the probabilities for this sentence belonging to a specific class are:

P(Cpos) = 0.141 \* 50 / 240 \* 10 / 240 \*100 / 240 \* 70 / 240 = 1.49 \* 10^-3

P(Cneu) = 0.723 \* 500 / 1230 \* 90 / 1230 \* 600 / 1230 \* 20 / 1230 = 1.71 \* 10^-4

P(Cneg) = 0.141 \* 50 / 240 \* 10 / 240 \* 100 / 240 \* 10 / 240 = 2.12 \* 10^-4

This sentence can thus be classified in the positive category.

Results

To test the Naïve Bayesian classifier, first, I used a sample of 15000 restaurant reviews dataset extracted from the filtered Yelp Challenge Dataset. I also split this dataset in 80% for training and 20% for testing. As shown in Table below, the best accuracy score on test dataset (3000 reviews) was achieved by the system (69.36%) using the approach when punctuations were removed, n-grams used, no stemming performed and stop words not removed in forming the feature vectors in bag of words representation.

Second, I used a sample of 6 000 restaurant reviews dataset extracted from the filtered Yelp Challenge Dataset. I also split this dataset in 80% for training and 20% for testing. As shown in Table below, the best accuracy score on test dataset (1500 reviews) was achieved by the system (67.33%) using the approach when punctuations were removed, n-grams not used, no stemming performed and stop words not removed in forming the feature vectors in bag of words representation.

Above statistics indicated that as the size of the training dataset was increased the accuracy of the classification was improved.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Training/Test Ratio (80/20 => 12000/3000 reviews)** | | | **Training/Test Ratio (80/20 => 6000/1500 reviews)** | | |
| **Features** | without removing stop words  + removing  punctuations + without stemming + without n-grams | Without removing stop words  + removing  punctuations + without stemming + with n-grams | removing stop words  + removing  punctuations + stemming + with n-grams | without removing stop words  + removing  punctuations + without stemming + without n-grams | Without removing stop words  + removing  punctuations + without stemming + with n-grams | removing stop words  + removing  punctuations + stemming + with n-grams |
| **Accuracy** | 67.96 | 69.36 | 66.89 | 67.33 | 66.66 | 65.94 |

Table 3: Accuracy results for Naïve Bayesian

Above statistics indicated that as the size of the training dataset was increased the accuracy of the classification was improved. Figure 1 represents the changes in accuracy depending on different features used for two different training data sizes.

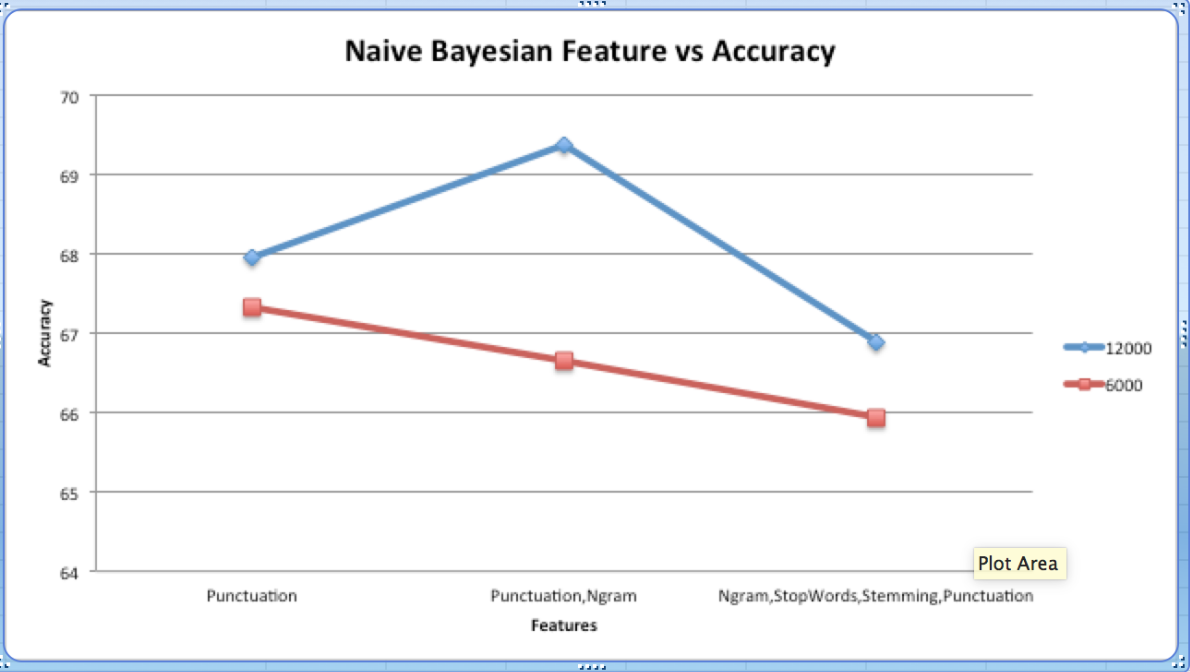
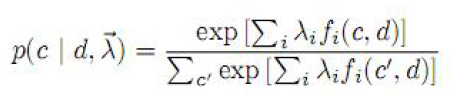


Figure 1. Naïve Bayesian – Feature vs Accuracy

**Maximum entropy Model**

The idea behind Maximum Entropy classifiers is that we should prefer the most uniform models that satisfy any given constraint. Maximum Entropy models are feature based models. We use these features to find a distribution over the different classes using logistic regression. The probability of a particular data point belonging to a particular class is calculated as follows:



Where, c is the class, d is the data point we are looking at, and λ is a weight vector. MaxEnt makes no independence assumptions for its features, unlike Naïve Bayes. This means we can add features like bigrams and phrases to MaxEnt without worrying about feature overlapping. This project uses NTLK for MaxEnt implementation.

Results

To test the Maximum Entropy classifier, first, I used a sample of 15000 restaurant reviews dataset extracted from the filtered Yelp Challenge Dataset. I also split this dataset in 80% for training and 20% for testing. As shown in Table below, the best accuracy score on test dataset (3000 reviews) was achieved by the system (64.39%) using the approach when punctuations were removed, n-grams used, stemming performed and stop words removed in forming the feature vectors in bag of words representation.

Second, I used a sample of 6 000 restaurant reviews dataset extracted from the filtered Yelp Challenge Dataset. I also split this dataset in 80% for training and 20% for testing. As shown in Table below, the best accuracy score on test dataset (1500 reviews) was achieved by the system (67.33%) using the approach when punctuations were removed, n-grams not used, no stemming performed and stop words not removed in forming the feature vectors in bag of words representation.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Training/Test Ratio (80/20 => 12000/3000 reviews)** | | | **Training/Test Ratio (80/20 => 6000/1500 reviews)** | | |
| **Features** | without removing stop words  + removing  punctuations+ without stemming + without n-grams | Without removing stop words  + removing  punctuations+ without stemming + with n-grams | removing stop words  + removing  punctuations + with stemming + with n-grams | without removing stop words  + removing  punctuations+ without stemming + without n-grams | Without removing stop words  + removing  punctuations+ without stemming + with n-grams | removing stop words  + removing  punctuations+ stemming + with n-grams |
| **Accuracy** | 63.95 | 64.19 | 64.39 | 62.13 | 63.26 | 62.4 |

Table 4: Accuracy results for Maximum Entropy

Above statistics indicated that as the size of the training dataset was increased the accuracy of the classification was improved. Figure 1 represents the changes in accuracy depending on different features used for two different data sizes.

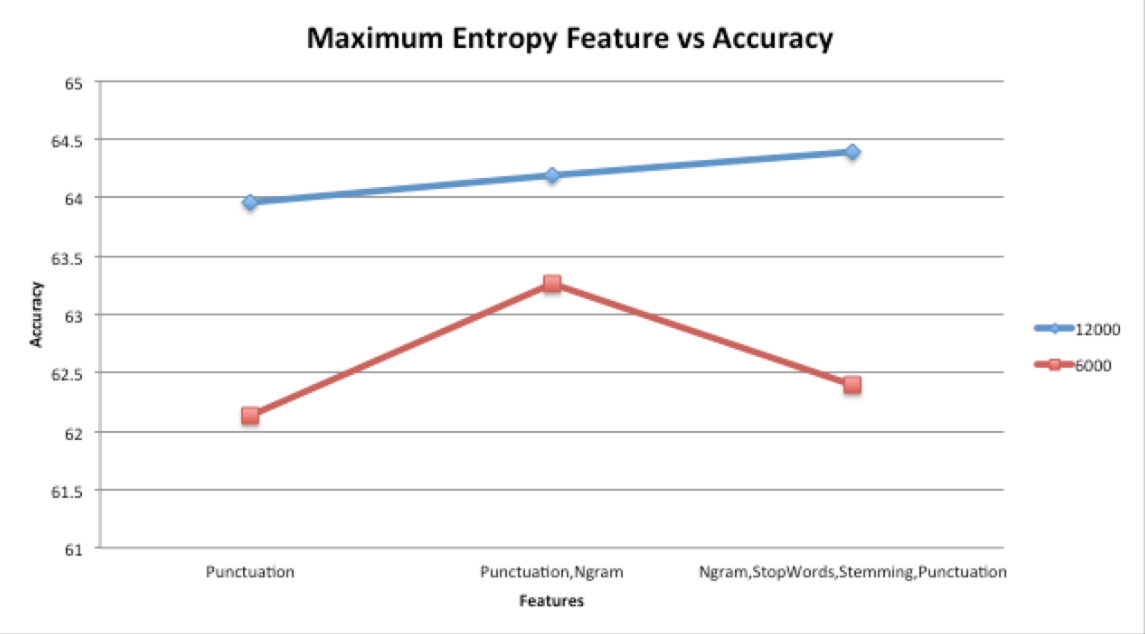


Figure 2. Naïve Bayesian – Feature vs Accuracy

**Restaurant Tagging**

After performing the restaurant review classification as positive/negative/neutral, the next performed was rating the restaurant as Good/Average/Bad based on the percentages of positive, negative and neutral reviews for that restaurant.

For this testing, review data for 3 restaurants was collected. The restaurants chosen were “Panera Bread”, “P.F. Chang's” and “Applebee’s”. Out of the two classifiers used, Naïve Bayesian classifier produced best accuracy using the approach when punctuations were removed, n-grams not used, no stemming performed and stop words not removed in forming the feature vectors in bag of words representation. So, using this approach the test data of these three restaurants was run, which provided the predicted tones of the reviews for these restaurants.

After this depending on the number of positive, negative or neutral tone reviews for the restaurant, it was labeled as Good, Average or Bad.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Restaurant** | **Positive** | **Negative** | **Neutral** | **Tag** |
| Panera Bread | 6 | 6 | 10 | **Average** |
| P.F. Chang's | 31 | 26 | 29 | **Good** |
| Applebee’s | 4 | 1 | 2 | **Good** |

Table 5: Restaurant Tagging results based on predicted review tones

**V. Conclusion**

This work has demonstrated the difficulty of extracting relevant sentiment information from yelp dataset restaurant reviews and identifying the tone of user review as positive/negative/neutral which in turn is used as a predictor for restaurant rating. This work explores the usage of feature extraction methods and classifiers like Naïve Bayes and Maximum Entropy Model for classifying restaurant text reviews using a large dataset.

For this project, the Naïve Bayesian was the best classifier and has obtained an accuracy of 69.36% using the Bag of words approach proposed in the feature extraction algorithm. In terms of performance, Maximum Entropy classifier tends to have slightly worst results. In case of both the classifiers, I feel that the use of n-gram approach that I followed affected the accuracy when it was coupled with stop words removal and stemming. It did not perform as expected, resulting in the less accurate classification.

For future work, increasing the size of training data and trying out some other classifiers like Linear SVM, SGD etc. can help improve the accuracy of classification. Also, the system’s accuracy could still be improved by exploring the usage of part-of speech (POS) as features in order to distinguish between the same word features that are used as different POS.

REFERENCES

[1] [Online]. Available: <https://www.yelp.com/dataset_challenge/dataset>

[2] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up?: sentiment classification using machine learning techniques," Proceedings of the ACL-02 conference on Empirical methods in natural language processing, vol. 10, pp. 79-86, 2002

[3] X. Yun, X. Wu and Q. Wang, "Sentiment Analysis of Yelp’s Ratings Based on Text Reviews", pp. 1-5, 2014. [Online]. Available: <http://cs229.stanford.edu/projects2014.html>

[4] J. Jong, "Predicting Rating with Sentiment Analysis," pp. 1-5, 2011. [Online]. Available: <http://cs229.stanford.edu/proj2011/JongPredictingRatingwithSentimentAnalysis.pdf>