Predicting Yelp Ratings using textual analysis

Neil Sengupta

10^{th} December, 2015

Abstract

This paper analyzes the data set from the Yelp data set challenge. The first section talks about the features, characteristics and statistics of the Yelp data set. It also provides information about some patterns within the data set. The next sections identify the problem of predicting Yelp ratings based on textual information and the training and evaluation of different models to perform the predictive task. The goal of this paper is to predict ratings using textual information and at the same time perform sentiment analysis with those words.

1 Introduction

Yelp helps people find good quality local businesses like restaurants, coffee shops. The Yelp data set in turn has the following characteristics- 1.6M reviews and 500K tips by 366K users for 61K businesses, 481K business attributes, e.g., hours, parking availability, ambiance.Social network of 366K users for a total of 2.9M social edges and aggregated check-ins over time for each of the 61K businesses.

2 Data Source

The data set used is a subset of the Yelp data set which is publicly available(http://www.yelp.com/dataset challenge). The data provides important textual and review information about a wide variety of different operating businesses. The data set provides information like type of business, actual textual review, user profile, star ratings/votes and user profiles.For the purpose of this report, we use the files yelp_academic_dataset_business.json and yelp_academic_dataset_review.json. The data set consisting of reviews is presented in the form of a json file which has the following format:

```
{
 type: review, 1
 business_id: (encrypted business id),
 user_id: (encrypted user id),
 stars: (star rating, rounded to half-stars),
 text: (review text),
 date: (date, formatted like 2012-03-14),
 votes: {(vote type): (count)},
}
```

Initially all the reviews were being considered for training the model, but then the scope was reduced to only choosing reviews that are in English, since it would be tedious to train a model for a different language. Thus, all of the reviews being used for training the model is in English and the user base has been capped to only those from the United States. In order to reduce the complexity for training the model, the data set was reduced to include reviews of only restaurants from six states in the United States - Charlotte, Las Vegas, Madison, Phoenix, Pittsburgh and Urbana-Champaign. The information was extracted from the json file into city_name.txt files. Each of the txt files contains the reviews and information for businesses that are located in the respective city. 90% of the data is used to train the model and the rest is used for testing the correctness of the model. The following table gives a preliminary statistic for the number of reviews from each state.

City Name	Number of Reviews
Charlotte	65855
Las Vegas	405760
Madison	30852
Phoenix	381614
Pittsburgh	46569
Urbana Champaign	8353

3 Motivation

Yelp usually has a lot of reviews about every operating business.Reading all the reviews takes up a lot of time to form an opinion. There is also a high probability that the different reviews have common words, expressions. The goal is to pick out the top occurring words for a set of reviews for a business, and understand it's context in determining the review of a restaurant/business. This method is straightforward, more comparative and saves more time. Then ngrams were constructed from the textual review, and a feature matrix was constructed, that contained the top occurring ngram with occurence. This was then fed to the training model. Most of the code for data processing was done with online, pre-written/library code.

4 Data Analysis Pipeline

The basic idea is to extract textual information and feed it into a model which can be trained and tested to spit out ratings solely based on textual data possibly by analysing word frequency and/or context. So first, the reviews for the six states were extracted into a text file. Then an online library was used to remove unnecessary words/unmeaningful words

from the text. After that word stemming was used, that uses the Porter streaming algorithm retrieve the word in it's base/root form. Stopwords included in nltk.corpus were removed from the text.

5 Prediction Model

The goal is to accomplish a predictive task of generating the review of a restaurant based on textual extraction of reviews of them. This problem is tackled by training a linear regressor for every state. The feature space here consists of the top 1000-3000 occuring unigrams/bigrams for every state. The model extracts these n-grams from each of the .txt file and tries to predict the review for that respective restaurant. The performance of the model is then judged in terms of testing and training error. 90% of the data set we have is used to train the regressor model and the rest of it is used for testing to determine the accuracy. We use the *Root Mean Square error* metric in order to compare and evaluate different models. This metric calculates the standard deviation between the predicted and observed values and produces a single predictive power output. The Root Mean Square Error is calculated using the following formulae: $\sqrt{\frac{\sum_{n=1}^{n}(y_i-\hat{y}_i)^2}{n}}$ A linear[1] regression model was used to predict the ratings using the text information. The model ouputs what words have negative and positive impact based on the value of θ(estimator). Also, training a linear regressor is much easier and was thus used for this prediction task. The txt file contains a bunch of words we extracted from the reviews. The top 1000 n(1)-grams are fed into the model to produce a rating. The more the number of n-grams are fed into the model, the better and more accurate is the output rating. Using this as the base case, different models were tested and compared. Here is a general algorithm of training the linear regressor and producing a rating:

for (Review text:training set) **do**
Make a dictionary of unique unigrams related to their frequency in the review text

Sort Dictionary and get top 1000 occuring unigram.

Def. feature function that uses a review and returns a feature vector of the corresponding occurences of those 1000 frequent unigrams and the feature vector length as offset.

Train the linear regressor using rating and feature_vector [count of review texts * 1001] Use testing set to analyze accuracy (calucalate) error of model.
end for

6 Results

The linear regression model was used to predict the ratings for restaurants in six different states. The training error and testing error was relatively the same for each of the states. Here is a table showing the training and testing error for Charlotte, North Carolina: The performance of the linear regressor is improved by using unigrams over bigrams. When we use more frequently occurring word, the training and testing error both go down for the model.

	Training Error	Testing Error
Unigram	0.8749	0.9279
Bigram	0.9927	1.0562
2000 Unigram	0.7139	0.9213
2000 Bigram	0.9461	1.0230
3000 Unigram	0.6869	0.9206
3000 Bigram	0.9201	1.0114

The average unigram testing error was a little over 0.9 whereas it was between 1.0 and 1.1 for a bigram.

- Some of the top positive words used in Charlotte, South Carolina are - Amazing, Fantastic, Awesome, Great, Fabulous, Excellent, Reasonable, Wonderful, Beat, Thank, Incredible, outstanding, reasonably, perfection, delicious, highly, die, favorite, perfect.

- Some of the top positive words used in Pittsburgh are - Favorite, amazing, awesome, incredible,gem, fantastic, excellent, reasonably, best, love, delicious, glad, wonderful, fabulous, outstanding, perfection, die.

- Some of the top positive words used in Pittsburgh are - Die, glad, best, ahead, perfect, recommended, best, amazing, ends, delicious, ahead, California, incredible, fantastic, yea, gotten, world, reasonable.

- Some of the top negative words used in Las Vegas are - mediocre, soggy, closed, dirty, worst, slow, lacking, overpriced, stay, oil, disappointing, horribly, bland, poor, greased, rude, sorry.

- Some of the top negative words used in Charlotte, NC are - dry,mediocre, terrible, bland, cold, frozen,worst, money, awful, dirty, disappointing, horrible, sad, rude, slow, sorry, overpriced.

- Words frequently used in Pennsylvania restaurant reviews - horrible, worst, terrible, mediocre, awful, rude, meh, unfortunately, avoid, poor, disappointing, bland, dirty, overpriced, barely, sad, money.

As we can see, that most of the reviews actually have matching words or overlapping words. Thus we can run sentiment analysis using words and their contextual information to produce a rating for a restaurant.

7 Observations and Conclusions

It is noticed that increasing the number of more frequently occuring words to train our regression model improves it's output. For example, in the North Carolina training set there are 77737 unigrams. Thus, having more and unique unigrams will improve the result of the model. However, processing a large number of words will also cause the computation time to fairly increase. Testing shows that using a unigram, the training and testing error is reduced. To process bigrams we require a better text-mining and context analysis model. There 1500000 unique bigrams, and thus using a couple thousand bigram words may not be sufficient for text analysis. Term frequencyinverse document frequency was additionally tested in order to weight the importance of the unigrams with respect to the rest of the review. The word count was thus added to the feature space when training the model. Upon testing using the top 3000 unigrams the training and testing error turned out 0.874886 and 0.927893 respectively.This shows that using this method we didn't get much of an optimization, because it only picks out weighted important words from the text. Thus just using unigrams from the textual reviews, the linear regressor is able to produce acceptable ratings.

8 Optimizations

The current model to rate restaurants produce reasonable and fairly accurate results. However, there is room for optimization. By improving and changing the feature set fed into the model, one can expect lower error rates on testing. Theoretically, the following optimizations are possible on the data analysis pipeline:

a)Better algorithms to perform sentiment analysis on text(Extra Trees Classifier, Gradient Boosting Classifier, Random Forest Classifier, Logistic Regression etc).

b)Include more features to train the linear regressor(word count, word occurence, context).

c)The n-gram model can be improved by increasing the count of frequently occuring words and implementing computational linguistics algorithms like grammatical tagging(POS-taggers), parsers for natural languages etc.

9 Bibliography

[1]Predicting a Business Star in Yelp from Its Reviews Text Alone by Mingming Fan and Maryam Khademi

2)Prediction of Yelp Review Star Rating using Sentiment Analysis by Chen Li (Stanford EE) Jin Zhang (Stanford CEE)

3)CS 229 Machine Learning Final Projects, Autumn 2014

4)CS229 Project report:Yelp Personalized Reviews by Thomas Palomares, Alexis Weill, Arnaud Guille

5)http://aimotion.blogspot.com/2011/10/machine-learning-with-python-linear.html

6)Exploring the Yelp Data Set: Extracting Useful Features with Text Mining and Exploring Regression

7)http://www.runzemc.com/2014/12/predicting-yelp-ratings-using-textual-reviews.html