Text Mining for Yelp Dataset Challenge

Mingshan Wang
A98099513
Email: miw034@ucsd.edu

Ruiqing Qiu
A98022702
Email: rqiu@ucsd.edu

Abstract

For assignment 2, we analyze an interesting dataset that a lot of research is conducted on : Yelp Dataset Challenge. This challenge dataset is basically a huge social network of 366K users for a total of 2.9M social edges. In the first section, we will describe more details about the basic statistics and properties of the dataset, and report some interesting findings in this dataset. In the next section we will identify a predictive task and utilize different models we learned in class to accomplish the predictive task. In the third section, we will graph some results to illustrate different models' performance. Here we will evaluate our model to using RMSE. In the forth section, we will further discuss about our results. And later on we will talk about some related work and future work. Our goal of this project is to use only review text to predict the rating, at the same time find out what are some positively related words, and what are some negatively related words in this dataset's review texts.

1 Basic statistics and properties

Yelp challenge dataset contains information about local businesses, reviews and users in 10 cities across 4 countries. There are 5 json files provided in this dataset. Since we are interested in how review text can be used to predict rating and business dataset contains rich information about the characteristics of the business. Therefore, we make use of the two out of five json files:

- **yelp_academic_dataset_business.json**: (55.4MB) type,bussiness_id, name,neighborhoods,full_address, city,state,latitude,longitude,stars,review_count, categories,open, hours,attributes
- **yelp_academic_dataset_review.json**: (1.43GB) type,business_id,user_id,stars,text,date,votes

Initially, we started using the whole dataset and try to train a model from there. However, we found that reviews from Germany are in German, which means we can't train a English model with a German model at the same time. Therefore, we decided to only includes those review texts within United States to be our target. However, the dataset is still huge and training one linear regressor took us a long time. Then, we decided to partition the dataset into six different states and only targeting those that are considered as restaurant. All reviews contain the business id where then we can retrieve the location of the business. There are 6 different cites in the United States that the dataset consists of: **Pittsburgh, Charlotte, Urbana-Champaign, Phoenix, Las Vegas, Madison**. We split the review texts according to different states. After we preprocess the json files, the number of reviews in each state is listed in Table 1.

City Name	State Name	No.of Reviews
Phoenix	AZ	381614
Pittsburgh	PA	46569
Charlotte	NC	65855
Urbana-Champaign	IL	8353
Madison	WI	30852
Las Vegas	NV	405760

Table 1: Number of Reviews Distribution Among Different States

- **NC.txt & NC_test.txt**: Dataset containing the review information for businesses that are located in Charlotte, NC. 90% from the original dataset are our training data and 10% will be our test data to evaluate our models
- **All Other 5 states**: Same as above for the rest of 5 states

Our motivation to predict rating using purely review text is that although Yelp reviews are really valuable, people normally won't have enough time to go through all the reviews when they deciding which restaurant they want to go. Therefore if we can find a way to capture the most important and meaningful words in the review text and use those words to predict the restaurant rating, that will save users a lot of time because rating is more comparative and straight-forward.

We plot some graphs to show the basic statistic of the dataset. Fig. 1 shows the most frequent words among all restaurant reviews in Charlotte, North Carolina. Fig. 2 shows the distribution of review length in our dataset.

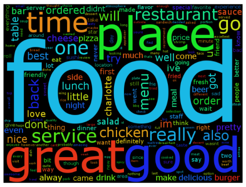

Figure 1: Most frequent words in all restaurant reviews in Charlotte, North Carolina[1]

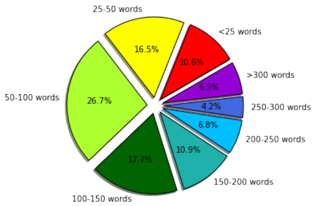

Figure 2: review length for all restaurant reviews in Charlotte, North Carolina

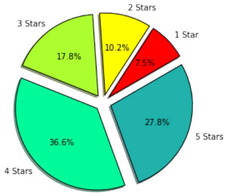

Figure 3: distribution of review stars in Charlotte, North Carolina

In Fig 4. the red dots represent all the restaurant location within the U.S.

2 Predictive Task

Our predictive task is to predict Yelp rating solely based on review texts. Our model is to train six linear regressors for each state. The feature space is the most 1000/2000/3000 frequent occurring unigrams/bigrams in that state's review text plus an offset, and compare the performance in terms of training error and test error. We split each state's review texts into training set and test set. Training set size is 90% of the total number of reviews and test set size is 10% of the total number of review texts for that state. That is, we have 6 training sets, 6 test sets. To compare and evaluate different models, we calculate the Root Mean Square Error (RMSE) on the train and test data to compare different methods. RMSE definitions are as follows:

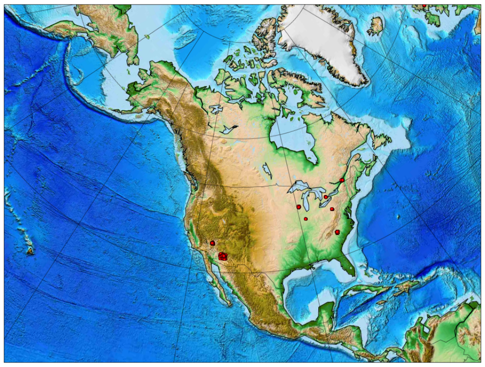

Figure 4: All the restaurant locations within the U.S.[2]

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n} (y_i - \widehat{y}_i)^2}{n}} \tag{1}$$

We choose RMSE because it is a good measure of accuracy, that is our model's predictive power. And it is better to use RMSE when we need to compare errors of different models for a particular variable, in our case is rating. So we adapt RMSE to our evaluation.

From this paper [3]. we learned that linear regressor performs the best, so we decide to use linear regressor as our model. In addition,linear regressor can give us what words are more positively/negatively related to the rating based on their theta values. What's more,linear regressor is relatively easier to train, so we believe linear regressor is the best fit for our predictive task.

Our baseline is using most 1000 frequent occurring unigrams to predict the rating. Our guessing is more words we include in our feature vector should improve the predictive result. Build up from the baseline, we can come up with different models and compare with the baseline.

Here we will use the most 1000 frequent occurring unigrams in Charlotte, North Carolina as an example to illustrate how do we process the data to get the feature and evaluate the linear regressor. The steps we take to train our linear regressor is as follow:

First Loop through all the review texts in the training set and have a dictionary of unique unigrams associated with their number of occurrences in all the review texts.

Second Sort the dictionary and get the most 1000 frequent occuring unigram

Third Define a feature function which takes a single review and return a feature vector of the corresponding occurences of those 1000 frequent unigram, plus an offset,that is the feature vector length is 1001.

Forth Use the star as the actual rating, along with the feature matrix of size [number of review texts * 1001] to get the linear regressor

Fifth Use our test set to evaluate the linear regressor performance

3 Results

We use our models to predict the rating in six different states, but the training errors and test errors are relatively similar for each state. So we will show the result of Charlotte, North Carolina as a typical example.

Table 2: Training Error and Test Error different model

	Train RMSE	Test RMSE
Unigram	**0.8749**	**0.9279**
Bigram	**0.9927**	**1.0562**
Unigram 2000	**0.7139**	**0.9213**
Bigram 2000	**0.9461**	**1.0230**
Unigram 3000	**0.6869**	**0.9206**
Bigram 3000	**0.9201**	**1.0114**

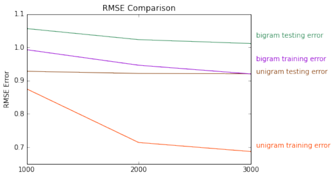

Figure 5: Unigram and Bigram traing/test RMSE error distritbution

From Table 2 and Fig. 5, we can tell that unigram generally does better job than bigram. In addition,when the number of frequent words we used as our feature space increases, the test error goes down a little bit but training error decreases a lot for unigram model. Fig. 6,Fig. 7 and Fig. 8 are some examples of positively related words we extracted from theata values. Those words are really reasonable and indeed capture the meaning in a positive review text.

And Fig. 9,Fig. 10 and Fig. 11 are some examples of negatively related words.

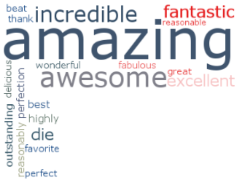

Figure 6: Top 20 positive words in all restaurant reviews in Charlotte, North Carolina

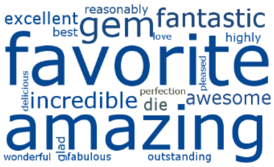

Figure 7: Top 20 positive words in all restaurant reviews in Pittsburgh, Pennsylvania

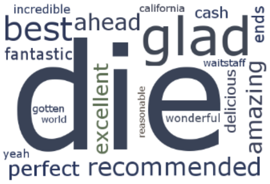

Figure 8: Top 20 positive words in all restaurant reviews in Las Vegas, Nevada

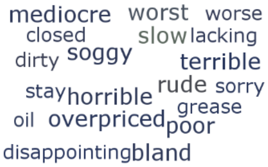

Figure 9: Top 20 negative words in all restaurant reviews in Las Vegas, Nevada

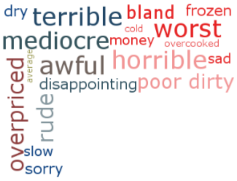

Figure 10: Top 20 negative words in all restaurant reviews in Charlotte, North Carolina

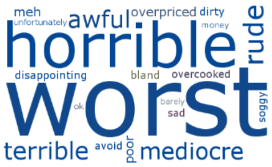

Figure 11: Most frequent words in all restaurant reviews in Pittsburgh, Pennsylvania

4 Discussion

The reason why more number of frequent occurring words we are using for our feature space gives better result is that there are **77731** unique unigrams in the Charlotte, North Carolina training set. Due to the large number of unique unigrams, we believe more unigrams will have a better result. But there is also a trade-off between running time and number of most frequent words we want to use.

In addition, similar to our result in homework 4, unigram gives lower training and test error than bigram. The reason might be that we didn't consider part-of-speech and other text mining approach to capture the meaning of the bigrams. What's more, there are **1509203** number of unique bigrams, 3000 bigram words might be not enough to capture the most important bigrams.

In addition to use word count as our feature space, we wonder whether using TF-IDF instead would help. Therefore, we did an addition experiment which we use most common 1000 unigram and TF-IDF as their feature space. The training error is 0.874866942188 and test error is 0.927892100169, which is extremely similar to the previous result where we use word count as our feature. This result contradicts to what we expect. When we print out the feature matrix, we find out most of the non-zero values are negative. The interpretation about that is those top 1000 words all appear in so many documents, so their TF-IDF value is pretty low.

In conclusion, TF-IDF doesn't improve much compared to word count as our feature. TF-IDF finds the most relevant words in the review text but those words doesn't help improving prediction on rating. Instead, using unigram has stronger predictive power than using bigram in general and the testing error of unigram linear regressor is pretty low. From our results, we can conclude that Yelp's review text can be used to predict rating very well by simply a bag of words model.

5 Related Work

Currently, researchers are trying to corporate *sentiment analysis* and *opinion mining* when using review text to do predictive task. The definition of *opinion mining* is a process for tracking the mood of the public about a certain product. *Sentiment analysis* refers to various methods of examing and processing the data in order to identify a subjective response, usually a good mood or a group's opinions about a specific topic. Broadly, *sentiment analysis* and *opinion mining* denote the same field of study. Finding the subjective meaning inside a Yelp review text will help a lot in determining the accurate rating of one business.

6 Future Works

There are still a lot of possible ways to improve our model to gain better predictive result. Here are some of them we can think of:

1. To improve bigram model, one can increase the number of frequent accruing word as the feature, or trying to do some trick such as part-of-speech tagging to capture the subjectivity of bigrams.

2. Include additional features in the linear regressor, such as the number of words in a single review text.

3. Try various ways of sentiment analysis, it should help predicting the rating.

References

[1] AMUELLER. How to generate word cloud using python.

[2] EHMATTHES. Visualization: Mapping global earthquake activity.

[3] FAN, M., AND KHADEMI, M. Predicting a business star in yelp from its reviews text alone.

7 Bonus Images

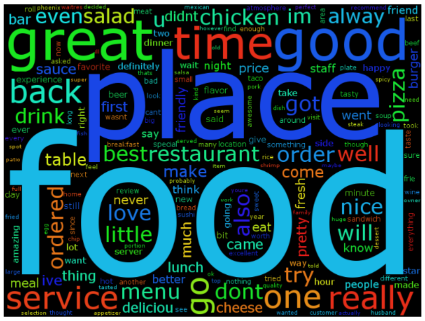

Figure 12: Most frequent words in all restaurant reviews in Phoenix, Arizona

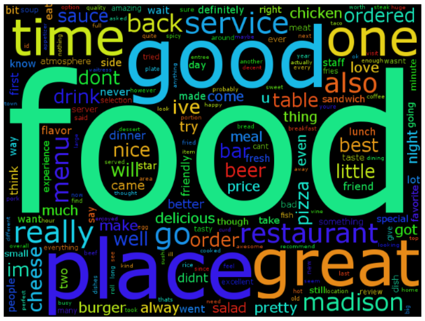

Figure 13: Most frequent words in all restaurant reviews in Urbana-Champaign, Illinois

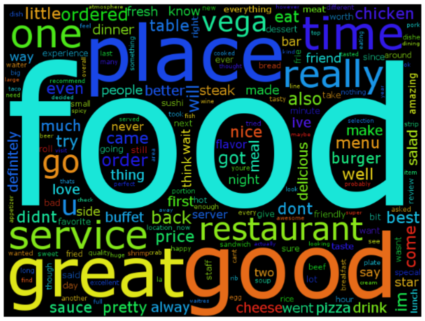

Figure 14: Most frequent words in all restaurant reviews in Las Vegas, Nevada

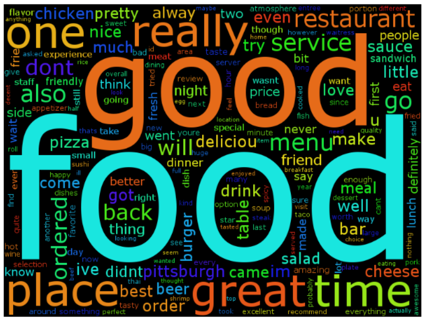

Figure 15: Most frequent words in all restaurant reviews in Pittsburgh, Pennsylvania

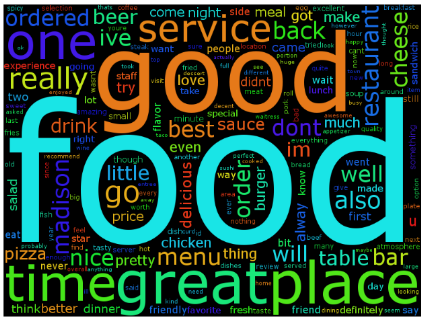

Figure 16: Most frequent words in all restaurant reviews in Madison, Wisconsin