CSE 190 Assignment 2

Jiaxi Chen
jic133@ucsd.edu
A99021050

Shuo Huang
shh093@ucsd.edu
A99001350

Pei Han
phan@ucsd.edu
A99049832

Abstract—As the world's biggest website for rating and commenting on local businesses, Yelp does not only just provide people with convenience, but more importantly it collects a huge amount of data for people to study. However, the sheer amount of ratings or comments can be somewhat too overwhelming for people to use. Sometimes Yelpers may write a lot for a restaurant but give a low rating to it, or sometimes they write few words but give 5 stars. In order to investigate the relationship between Yelpers' comments and ratings, we want to build a model which predicts ratings based on analyzing the words in each comment. Basically, our team have tried 3 approaches to decide word lists for model training and 2 ways to transform word frequency into feature vector for training, and then we use root mean square error (RMSE) to evaluate the performance of these models.

1. Introduction

With the sharp upsurge in people's passion for smart phones these years, web-based platforms become more popular than ever. Yelp, as the world's pioneering Internet Corporation, benefits millions of people with their everyday lives. People all over the world use Yelp to decide what to eat, at which hotel to stay, or even at which saloon to get a haircut. To make decisions, for example, Yelp users can compare the attributes of their targeting restaurants such as distance or price with others. Among all these attributes, our team think that user ratings and comments are the most critical and helpful ones that will lead users to make their final decisions. Therefore, our goal is to construct a model that can predict a rating precisely when given some user comments for a specific business.

Here we are using the data from Recsys Challenge 2013: Yelp business rating prediction [1], provided by Kaggle. In total, we have 229907 sets of reviews for 11537 businesses. Each review

has several attributes, including user/review id, stars, text, date, etc. Some important attributes of business data are business id, categories of business, location of businesses. The reviews are given by 43873 users so that the data are pretty representative.

In studying this dataset, we find some interesting fact that although the location and categories of business may tell us some information about how this business would be rated, given the models we learned from the class, it's still difficult to predict the rating of this business simply with attributes of them. Instead, we intend to do text analysis on reviews to help predicting the rating of each business.

2. Predictive Task

People have spent tremendous efforts on mining data on web and building more powerful model for analyzing reviews. Currently more and more contemporary researches start to focus on mining reviewers' opinions. Some studies have tried to analyze the product feature based on nature of human language [2], and other studies try to use the sentiments of wordings to construct features [3]. These works really help us establish our basic idea of how to design the prediction model.

As mentioned before, we are predicting the ratings of businesses according to the comments they receive from users. The baseline of our study is somewhat similar to that of reference [4]. In article [4], authors choose the k most used words, where each of the k popular words will end up having a specific weight, then they use the weights to train several models and test each of them on test data. Authors give the conclusion that linear regression has a better performance in predicting ratings (5, Fan and Khademi). However, we haven't learned most of the learning models that the authors used in their work. Therefore, we choose to trust their findings and only use linear regression model to predict ratings. Instead, we want to figure out if there are better ways to choose word list for training in addition to simply choosing the k most used words. Also, we try to examine if there is a better feature generation methods to transform word frequencies into feature vector. To measure how precise and reliable the models are, we use root mean square error (RMSE) as the standard metrics.

3. Methodology

A. Data

In paper [4], authors picked the businesses under largest category "restaurant" to work on. In our case, we choose the second largest category "nightlife" which has 30136 reviews as the data we mainly work on. This choice offers a large enough data to train test model on while the computation wouldn't take too much time. We randomly choose 26311 reviews as our train data. To train the model, we use train reviews and the rating they give to the business. To test the model, we use words in test reviews with filtering process illustrated by Fig 1 and train models to predict ratings and compare the results with the real rating of the businesses.

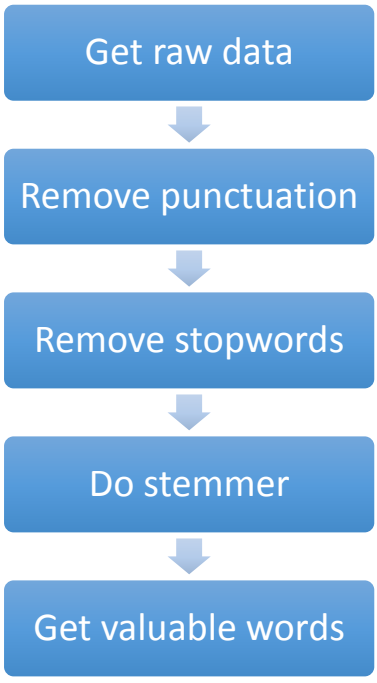

Fig 1. Process of Filtering Words in Reviews

Fig 2. 10 Most Used Words

B. Word List Selection

How well the model performs partly depends on the word list that we use to train. Since the data we have are all reviews to restaurants, clubs or bars, the most used words are "place", "good", "great", etc., as indicated by Fig 2. Some nouns do not represent and positive or negative attitudes and therefore have almost no use in prediction. What's more, there are three positive words in top 10 used words which are "good", "great", and "like", while no negative words showing up. It's possible that people tend to use more positive words than negative words in review, which make it harder for us to build rating predictor

based on sentiments of wordings. Therefore, we use the following three approaches to choose word lists with different k sizes including 30, 50, 100, 300, 500, 1000, 2000 and 3000.

1) Baseline: Top k used words

For the baseline, we simply choose the top k used words in all reviews under category "Nightlife".

2) Pos vs. Neg half and half

As indicated in Fig 2, the negative words are more likely to be neglected in baseline, compared with positive words. Actually, only 16% reviews have ratings less than 3 out of 5. Therefore, we choose top k/2 most used words in both review ratings larger or equal to 2 stars and review ratings less than 2. Eventually, we combine two parts into our final word list with size k.

3) Top k words with largest tf-idf value

We learned the concept of tf-idf (term frequency-inverse document frequency) in class:

$$tfidf(t,d) = f_{(t,d)} \times log_{10}\frac{N_d}{n_t}$$

In our case, term frequency refers to the number of times that term t appears in all reviews under category "Nightlife". N_d is the total number of reviews, while n_t refers to the number of reviews that term t appears no matter what category it belongs to. As shown by Fig 3, top 10 words with largest tf-idf value are able to represent "nightlife" better than Fig 2.

C. Feature Generation Method

1) Method 1

Count the number of times certain word appear in each review and use this value as the parameter value for this word in training model. Eg. Vector = [1, 2, 3, 0, 4, 0, 1] while the first "1" refers to the offset, the numbers in all the slots after refer to the number of times each word appears in the review.

2) Method 2

Put 1 as parameter value for certain word if this word is in the review we are working on, otherwise put 0 as parameter value. In other words, no matter how many times a certain word appears in the review, we only count it once. With the example I give for method 1, in method 2 the vector would be [1, 1, 1, 0, 1, 0, 1].

4. Result and Discussion

We now have three approaches to choose the words to be used as feature for linear regression formula. Also, we have two feature generation methods to transform words' frequency into

feature vector. Therefore, we try both frequency-to-feature methods on all three word choosing model and do comparison among their RMSE values. The result is represented by Fig 4-8.

First, let's consider values of k to create word lists. When k value increases from 30 to 1000, the RMSE decreases dramatically in all three word lists with two different feature generation methods. After k-value is 1000, the decrease of RMSE becomes much slower. Therefore, we can conclude that the Yelp rating predictor would have a good performance at a word list with size around 1000. The reason for it may be that when the value of k is small, we may neglect some negative words which have great impact on decreasing ratings. However, when the value of k is around 1000, most high-frequency negative words are included and therefore increasing size would not improve the model much but make the training more time-consuming.

Next, we look at the performance of all three work lists. We can observe in Fig 4-5 that with both feature generation methods, word list 3, given by picking top k words with largest tf-idf values, has the worst performance. That is, although the words in word list 3 have stronger

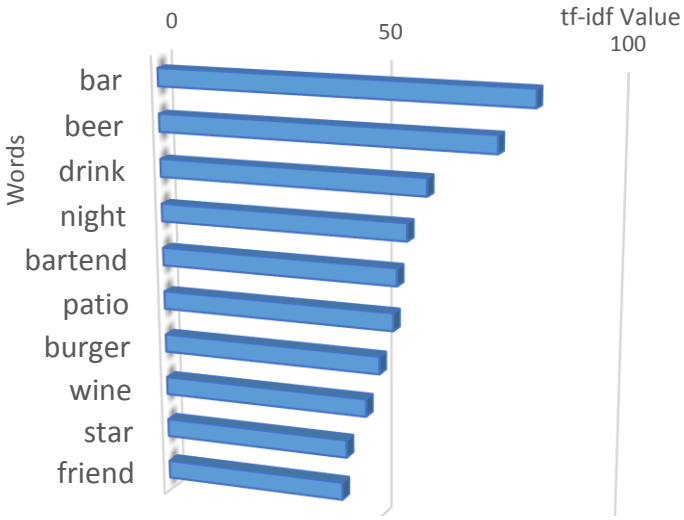

Fig 3. Top Words Related to "Nightlife"

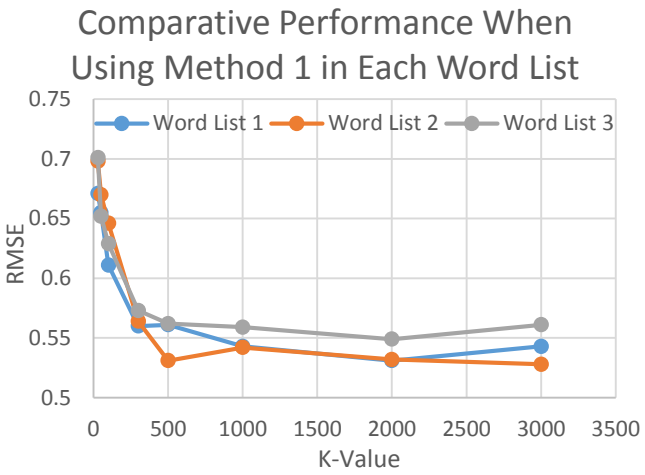

Fig 4. Performance of Feature Generation Method 1

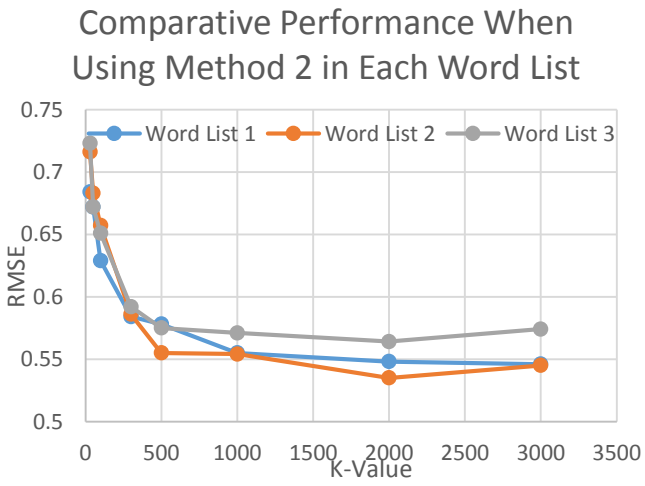

Fig 5. Performance of Feature Generation Method 2

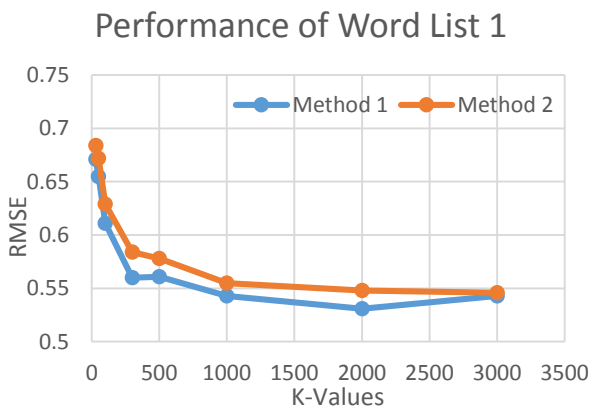

Fig 6. Performance of Word List 1

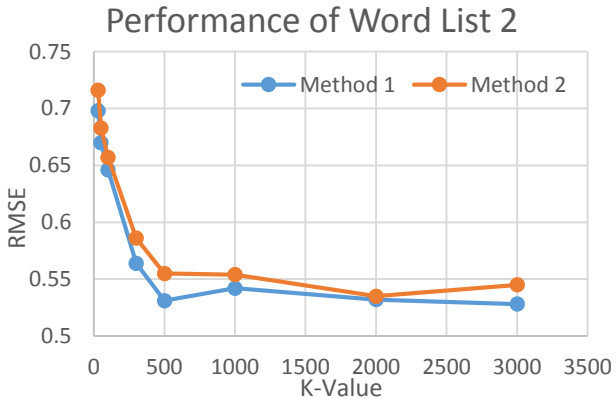

Fig 7. Performance of Word List 2

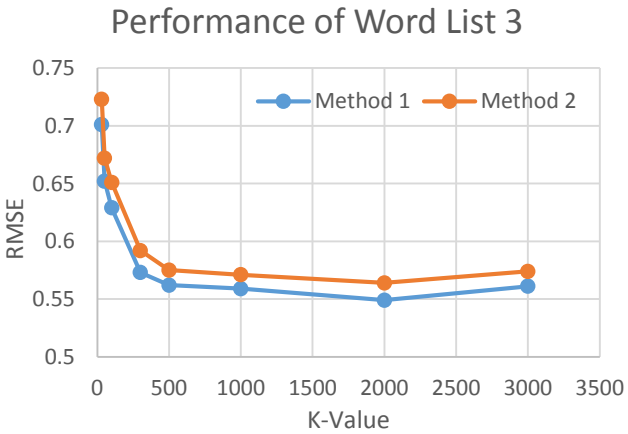

Fig 8. Performance of Word List 3

Feature Selection Method	K-Value	Average RMSE	
		Method 1	Method 2
Word List 1	≤ 300	0.624	0.642
	> 300	0.545	0.557
Word List 2	≤ 300	0.645	0.66
	> 300	0.533	0.547

Fig 9. Comparison of Two Word Lists

relationships with "Nightlife", they don't include enough words indicating reviewer's attitudes towards certain business. In other words, when eliminating the common words shared among different categories, we skip the common words people use to express their attitudes at the same time.

Word List 1 and Word List 2, however, perform at a similar level in both Fig 4-5. Therefore, we compute the average RMSE of these two lists in different ranges of k-value and displays the results for two lists with both feature generation methods in Fig 9. As we can see, no matter which generation methods we use, Word List 2 has a better performance when k-value is

larger than 300, while Word List 1, in contrast, performs better when k-value is small. As far as I'm concerned, one reason is that as suggested by Fig 2, negative words appear less frequently in the reviews compared with positive words and nouns. Even if we choose half of word list from reviews giving bad ratings, we cannot guarantee that much more negative words are added into word list. On the other hand, since the number of words we pick from reviews giving good ratings is also k/2, some positive words picked in baseline may also be neglected. As a result, the model we trained with Word List 2 performs worse than the one trained with Word List 1 when k-value is small. However, when k is close to the ideal value 1000 we discussed above, Word List 2 would include more negative words while retaining most of positive words in Word List 1, and as a result the model trained with Word List 2 has a better performance.

Lastly, let's see which feature generation method works better. In Fig 6-8, as we can see, method 1 always has a smaller RMSE than method 2. That is, the number of times each word appears in the review do affect the ratings users would give.

5. Conclusion

We tried three approaches to pack word list to be used to train linear regression model, which are appearance times, appearance frequency in all reviews, and tf-idf values. We have tested eight k values to see which top k-value as the size of word list can give a relatively reliable and efficient prediction. What's more, we have examined two feature generation methods to see which one would return a feature vector that tells reviewer's attitudes toward businesses better.

Comparing the results given by all possible combinations, we conclude that generating feature simply by the number of word's appearance (Method 1) is always better than generating feature by telling if a word appears in a review or not (Method 2). When k-value is less than 300, word list composed of top k most used words would result in a better model, while word list merged from two top k/2 most used words in bad reviews and good reviews respectively performs better when k-value is larger than 300. When predicting Yelp rating, word list with k-value around 1000 would give a relatively precise RMSE when other conditions are the same. Therefore, it's always better to use latter word list.

6. Future work

Although we find a relatively good model given specific k-value and word list selection, we still have a large space to improve out RMSE. One possible solution is to make the use of sentimental word list only counting the positive and negative words. This requires more efforts in the future.

7. Reference

[1] Kaggle.com,. 'Description - Recsys2013: Yelp Business Rating Prediction | Kaggle'. N.p., 2015. Web. 1 Dec. 2015. Available: https://www.kaggle.com/c/yelp-recsys-2013

[2] A.-M. Popescu and O. Etzioni, “Extracting product features and opinions from reviews,” in Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing, Stroudsburg, PA, USA, 2005, pp. 339–346.

[3] B. Pang and L. Lee, “Opinion Mining and Sentiment Analysis,” Found Trends Inf Retr, vol. 2, no. 1–2, pp. 1–135, Jan. 2008.

[4] Fan, Mingming, and Maryam Khademi. 'Predicting A Business’ Star In Yelp From Its Reviews’ Text Alone'. *arXiv.org*. N.p., 2015. Web. 1 Dec. 2015. Available: http://arxiv.org/pdf/1401.0864v1.pdf