Introduction and goal

An increasing number of people like to search for some information on the website when they have some plans. Yelp is one of the sources people always look for because it provides reviews written by customers of many aspects such as food, service, cost, etc. By reading the reviews, we can easily tell the feelings that customers convey and guess the ratings. On the contrary, it's difficult for machines to understand the sentiment of a text. Therefore, our goal is to figure out what makes a review positive or negative and propose a model to predict the rating mainly based on the text of the review.

Background information

In the training data, there are 1546379 records with 8 variables: stars (integers between 1 and 5), name, text, date, city (business location), longitude, latitude and categories. In the testing and validation data, there are 1016664 records with 8 variables. It has a variable called id which represents ID number for Kaggle instead of the stars variable in the training data. The remaining variable names are the same as the training data. Due to the huge dataset, it takes too long to standardize the text and adjust the parameters in a model. We just randomly pick up 30000 reviews in the training dataset and do sentiment analysis based on this data.

Data preparation

Variable selection

First, we look at the selected training data, finding that there is no missing value. Second, we draw a plot to see the distribution of stars which shows our sample is unbalanced. There are more reviews rated 4-5 stars than reviews rated 1-3 stars which means we tend to predict higher ratings for the business. Then we want to figure out the relation between stars and other variables.

1. Text variable must have something to do with stars and almost all the documents we have read reveal the model based on text variable alone. Also, we check the distributions of the length of characters for each star level that are slightly different but not a big deal.
2. City, longitude and latitude tell the same information about the business location. From the map, we can see that the points are gathered into three parts but the distribution of stars for each part looks quite similar. Thus, we can ignore these variables.
3. Categories may contain much information, but they don't have unified division. What's more, words in the categories are frequently mentioned in the text.
4. From the year 2005 to the year 2017, the number of reviews is increasing mainly because Yelp is more and more popular over the years with the widespread of the Internet. However, when we check the percentage of each star levels, 1-3 stars remain the same while the percentage of 4 stars decreases and percentage of 5 stars increases. Star distribution is stable over the months. Thus, there may be a relation between stars and years.

(All the plots can be found under the image folder.)

Text processing

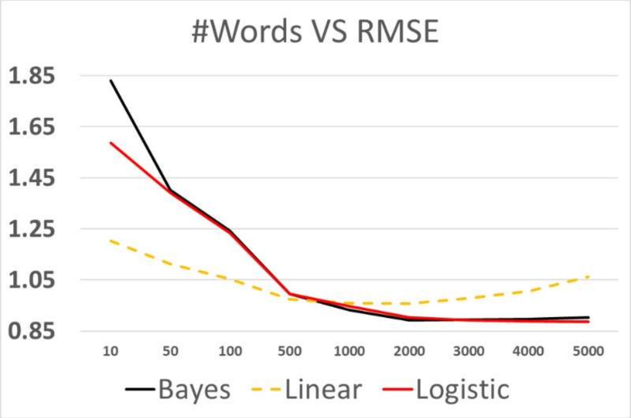
First, we look at the text variable and find that there are several texts written in the language other than English. Since the number is not large, we just ignore these texts. We use the average star level when we come across the other language texts in the prediction process. Then we remove noise from texts that include newline marks "\n", punctuation except "?" and "!" which may express the feelings. As for stop words, we keep some words like "her", "his" and some negative words. We keep 26 letters and change them into lowercase. Afterward, we do Lexicon normalization to change the tenses of verbs.

To do the feature engineering, we want to create a word dictionary to count the occurrence of every unique word. However, considering the word "never", "no", "not" followed by a word may change the meaning of one sentence, we combine these three words with the word right after it when building the word dictionary.

After creating the word dictionary, we notice that many words do not frequently appear with the count equal to one. In that case, we want to find a threshold to cut off the low-count words. After several trials, we decide to ignore the words whose count is less than ten. Then we start to select features. The histograms that show the relation between word and stars can tell a lot of information but we cannot go over 8000 histograms. Instead, we calculate the variance of stars for each word and select words with high variances. Before that, we do the following: scale the count of star levels for each word by the total counts, scale the count of star levels and scale function is log2 since the counts may differ a lot between words. As a result, we sort the words by their variance and select the top 1000 words. According to these words, we create a sparse matrix for the model. (You can check the example in presentation1.ppt to know how we deal with text.)

Model fitting

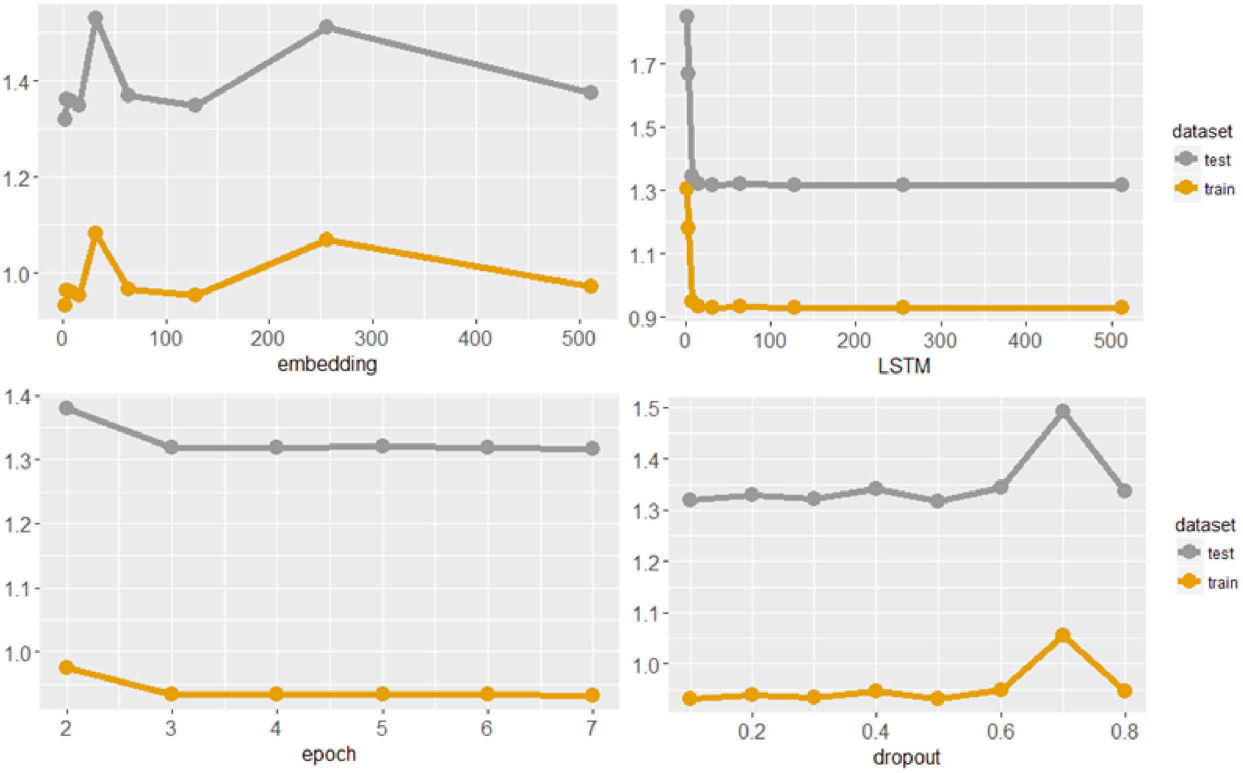
In the model fitting part, we try Linear Regression, Logistic Regression, Naïve Bayes and Deep Learning model. Deep Learning model gives the smallest RMSE which we use on the Kaggle, but we choose linear model for better interpretation. (We planned to add date variable but eventually find it makes the model worse, so we ignore it.)



Precise prediction

How to set up the model: From the literature, we learned that the word embedding method along with the long short time memory (LSTM) network is widely used on this topic.

Parameter tuning: In this part, we do a series of test with 10000 train data and 5000 test data. There are four parameters that matter a lot in this model: number of units in LSTM network, number of units in word embedding, number of epochs, dropout rate. We do tests with a series of different combinations of possible values. Finally we set 128 units LSTM network, 128 units in word embedding layer, 0.2 dropout rate, 4 epochs.



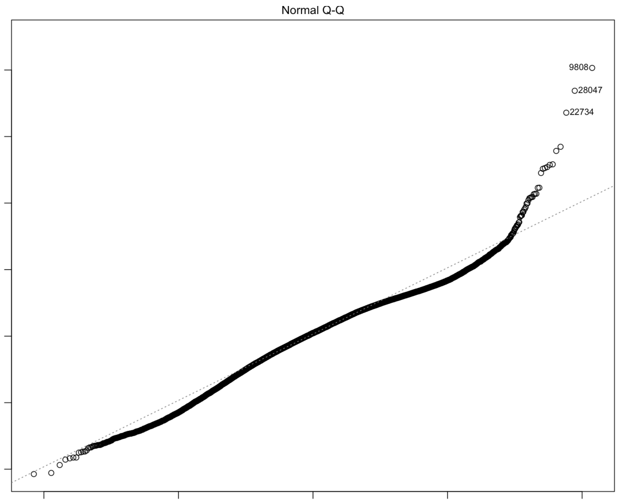
Final model: After the parameter tuning, we can set up our final model. With the parameters selected above, we train our model with the whole train data set with Keras. Finally, we achieve a score(RMSE) of 0.59889 on the test data.

Better interpretation

Based on the sparse matrix, we want to reduce the dimension so we consider principal component analysis. The first two principal components contain 96.75% information and the second principal component is the scores for each word. The scores are centered around zero so we choose words whose absolute value of score is larger than 0.5. In that case, we reduce 80% of the words so that have 1670 important words. The negative value of pc2 means the word is positive while the positive value of pc2 means the word is negative. By counting the positive and negative words in the review we can tell the feeling it conveys.

$$PC_1 = -0.434 \times S_1 - 0.45 \times S_2 - 0.45 \times S_3 - 0.45 \times S_4 - 0.444 \times S_5$$
$$PC_2 = 0.678 \times S_1 + 0.338 \times S_2 - 0.405 \times S_4 - 0.505 \times S_5$$

S_i represents the occurrence of word in star i. Then we do the linear regression based on 1670 words and its RMSE reaches 0.84. Among all words, we have 421 words with the significant p-value. All our variables explain 55% of the response variable. However, from qqplot we find the normality isn't satisfied well.



Strengths and weaknesses

Strengths

1. Deep Learning model gives precise prediction for ratings.
2. PCA separate positive and negative words effectively.
3. Linear Regression is easy to understand.

Weaknesses

1. When creating the word dictionary, we count every unique word and combine words like "never", "not", "no" with next word but ignore the other bigram.
2. We just ignore reviews written in languages other than English because the proportion is small. However, we think the best way to handle this problem is to translate other languages to English.
3. In order to propose the model more convenient and quick, we randomly select 30000 records which means we lose some information.

Conclusion

We first remove the noise in the text reviews and create a word matrix. Then we figure out what makes a review positive or negative using PCA method and give easy-follow interpretation by Linear Regression. Last, Deep Learning model gives the precise prediction rating.

Duties

Cheng Lu: Linear Regression, Logistic Regression, Random Forest and jupyter notebook summary.

Lan Wang: Naïve Bayes, SVM, PCA, presentation ppt and jupyter notebook summary.

Linhai Zhang: Text processing, Deep Learning model and jupyter notebook summary.

Reference

[1] Boya Yu, Jiaxu Zhou, Yi Zhang, Yunong Cao. Identifying Restaurant Features via Sentiment Analysis on Yelp Reviews.

[2] Mingming Fan, Maryam Khademi. Predicting a Business' Star in Yelp from Its Reviews' Text Alone.

[3] Mengqi Yu, Meng Xue, Wenjia Ouyang. Restaurants Review Star Prediction for Yelp Dataset.

[4] Nabiha Asghar. Yelp Dataset Challenge: Review Rating Prediction.

[5] Mingshan Wang, Ruiqing Qiu. Text Mining for Yelp Dataset Challenge.

[6] Yun Xu, Xinhui Wu, Qinxia Wang. Sentiment Analysis of Yelp's Ratings Based on Text Reviews.