Yelp Review Prediction

Yifan Li

December 3, 2018

1 Introduction

This project aims to:

1. find a explainable model to predict the score consumers will give (from 1 star to 5 stars) with their reviews.
2. find patterns of positive reviews (4-5 stars) and negative reviews (1-2 stars).
3. classify words into positive and negative.

Data I used is download from Yelp website. It contains about 1.5 million reviews with several columns. Only the *review* and *score* columns are used in this project.

The main idea of the model is to classify positive and negative words using the frequency ratios that each word appears in positive and negative reviews. And use the frequency of different types of words to classify positive and negative comments.

2 Method

2.1 Clean Data

Abbreviation and Special Symbols First, abbreviation and special symbols are modified. After the inspection, there exists eight situations.

Origin	n't	's	've	'd	'm	lve	cannot	\n
Now	not	is	have	would	am	I have	can not	<Space>

Here are some examples:

Origin	Now
Long story short.\n\nBunch ...	Long story short. Bunch
We didn’ t catch her ...	We did not catch her ...
It’ s super clean ...	It is super clean ...

Non-English Non-English reviews are removed. Here are some examples:

2ème arrêt au Barbù et nous avons vécu une,...
Corriente, sucio, y mal servicio. El cocinero,...
アリゾナ最後に良い旅の想い出の締めくくり,...
家から最寄りで食べに行けるベトナム料理...

They only account for one percent of the total reviews. So I directly delete them.

Negative Sentence The last and most important part is to deal with negative sentence. For examples, *brightest* is obvious a positive word. However, when it occurs in a sentence like ”there were not the brightest person”, it actually functions as a negative word. If we remain considering it as a positive one, it will ruins our prediction. For instance, words *well* and *return* which exists in the review ”the staff were obviously not well managed, I will never return to this location.” are both positive while this review brings no obvious negative word. Without special adjustment, this review will usually be taken as positive review in regression and given a high score. However, it is actually a one star review.

My idea is first finding words which makes a sentence as negative sentence. After checking several sentences, four words are found. They are *no, not, never, lack*. Sentences with these four words are usually negative sentence. I call them inverters. Then, for each negative sentence (sentences including inverters), I add a *not* in front of each word between inverters and the following punctuation. Here are some examples:

Origin	the staff were obviously not well managed, I will never return to this location
Now	the staff were obviously not notwell notmanaged, I will never notreturn notto notthis notlocation
Origin	Seriously can not stand this McDonald is. They never get my orderright.
Now	Seriously can not notstand notthis notMcDonald notis. They never notget notmy notordernotright.

Now, with *good* and *notgood*, we can determine whether it is in a positive sentence or negative sentence.

After doing all transformation, I delete all punctuation. It will lose some information. *!* or *?* may bring some special information. But I have no idea to analysis them. And I believe words is enough.

Due to the limitation of my computer, I randomly choose 100,000 reviews from total 1.5 million reviews to do further analysis.

2.2 Determine Positive and Negative Words

The most intuitive method to determine positive words is to rank words with their frequency of appearing in positive reviews (4 or 5 stars reviews). However, there is a big problem here. Some meaningless words appears significantly more frequent than other words. For example, the word *is* appears 6467 times among all 10,454 1 star reviews while the word *disrespectful* appears only 42 times. If we merely consider the frequency or equivalent probability, *is* will be taken as negative and *disrespectful* won't seem to be so negative. However, we know the opposite is true.

To avoid this kind of misclassification, probability ratio is used instead. I create five new variables named as Score1 to Score5. For example, the definition of Score1 for a word is:

$$\text{Score1[word]} = \frac{P(\text{this word is included in reviews with 1-star})}{P(\text{this word is included in reviews with stars other than 1})}$$

Definitions of Score2 to Score5 are similar, which just replace 1-star with 2-5 stars. Here are some examples to show the power of these five new Score variables.

Word	Variable	1-star	2-star	3-star	4-star	5-star
worst	frequence	1310	332	150	80	40
	probability	1e-3	4e-4	1e-4	3e-5	2e-5
	Score	18.3	1.9	0.5	0.107	0.038
notregret	frequence	7	5	10	70	177
	probability	8e-6	5e-6	8e-6	3e-5	8e-5
	Score	0.225	0.172	0.227	0.865	3.420
and	frequence	8943	8623	12542	25459	31314
	probability	0.0097	0.0093	0.0095	0.0111	0.0134
	Score	0.964	0.999	0.992	1.020	1.000

Table 1: Score of Words

From Table 1, we can see that although word *and* appears frequently, its distribution in reviews with stars is approximate uniform. And its Score1 to Score5 are closed to 1. However, for negative word *worst*, its Score1 is obviously larger than other scores. For positive word *notregret*, its Score1 is obviously

larger than other scores. (remember *not* in front of *regret* indicates that it is a negative sentence like "you will never regret". So that *notregret* is positive)

Here are two word cloud plot of words with high Score5 and Score1.

Figure 1: Positive

Figure 2: Negative

Here are some interesting findings. *die* is positive and *refund* is negative. I check some reviews which include these words.

1. i was told that our order would be **refunded**. as of the following tuesday it had not.
2. they refused to **refund** our money
3. the cake was so beautiful and absolutely to **die** for!!!!

When customers consider refund, it is not a good signal.

2.3 Determine Positive and Negative Reviews

After giving scores to words, now I need to give scores to reviews. Another five new variables S1 to S5 are created. For example, the definition of S1 of a review is:

$$\text{S1[review]} = \#\text{ of words with high Score1 in this review.}$$

After ranking by Score1, the first 1000 words are taken as "words with high Score1".

Definition of S2 to S5 is similar to S1. Now we can use these variables to do prediction.

3 LSTM model

Finally I fit a LSTM(Long Short-Term Memory) network of star$\sim$ S1+S2+S3+S4+S5. Details of my model could be found in "LSTM.ipynb" file. After cross validation, the total MSE is 0.63310782. Compared with least square lineal model, its MSE is about 1.1. It means LSTM network is a great improvement of lineal model.

Actually when I use the whole data set and take 80% as train data and rest 20% as test data, MSE of my model once reached 0.281. But due to the limitation of my computer, I can't do cross validation on the whole data. So I don't know whether there is any bias in splitting data.

4 Conclusion

1. Score1 to Score5 are powerful variables to determine a word positive or negative.

2. Although LSTM network is not interpretable, variables used in this model are interpretable.

3. My model is simple and basic but it still achieves good results. It means actually variable selection is more important than model selection in this situation.

4. There is no special reason to choose LSTM other than neural net model. So in the future, I could try other neural net model and do grid search over various model parameters to improve predictions.

5. Words with high Score1 and low Score1 have the same contribution on S1. Maybe I should try to allow different weights on them.

6. I find that negative reviews usually are longer than positive reviews. The length of reviews as a new variable may be helpful in prediction.