Yelp Review Prediction

1.1 Background Information and Thesis Statement

The main goals of this project are to identify a small set of informative features and a prediction model that manages to predict the ratings of reviews accurately. We also provide an interpretable model. Training data are about 1.5 million Yelp reviews and test data is 1 million.

1.2 Data Clean

- Modify Abbreviation and Special Symbol
  - Before: n’t
  - After: not

- Remove Non-English Reviews
- Negative Sentences
  - Before: They NEVER get my order right
  - After: they never notget notmy notorder notright

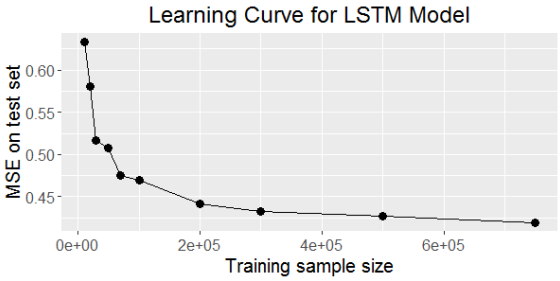
- Remove Punctuation

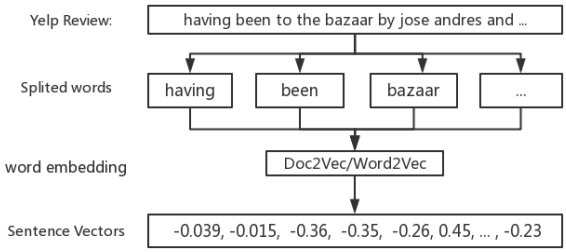
2 Model Description

Neural Network is a widely used method in text mining field. It provides more accurate predictions when predictors have complex patterns. And its calculation speed is much faster than other methods like SVM. We finally used Neural Network to build the prediction model on 500,000 samples. The model has 3 layers, a LSTM *(Long Short Term Memory)* layer with 50 output nodes followed by a Dense layer with 5 output nodes and then a Dense layer with 1 output node. The features used in the model consist of two parts:

- Pre-trained Sentence Vectors by word embedding process: Capture the word frequency and order information
- Additional Interpretable Variables directly generated from yelp data: Capture sentiment in the text as well as some date and location information

As the learning curve of the final model shown below, when the training sample size reaches 500,000, adding more training sample cannot cause obvious decrease on MSE of the test set.





2.1 Pre-trained Sentence Vectors

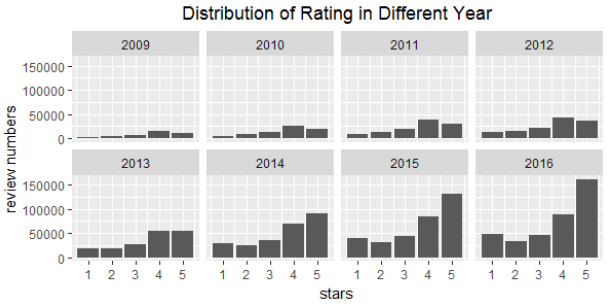
In order to capture the word counts information in the text with low dimension features, we used the word embedding process to convert each review into a 100-dimension vector. The word embedding is an unsupervised deep learning process. For the Word2Vec method, the main idea is to estimate words in nearby context. The embedding process uses Artificial Neural Networks. The input is list of sentences with separated words and the outputs are vectors for each word.

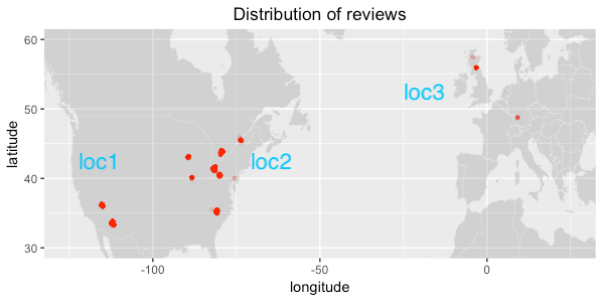
One way to convert a yelp review is to average the word vectors for each single word in the review, but it will again lose the order information in the sentence. Doc2Vec is a similar method as Word2Vec. The difference is that Doc2Vec can directly convert a whole paragraph or sentence to a vector without losing order information. We used the DBOW *(Distributed Bag of Words)* method in Doc2vec. The flow chart below shows the process to get sentence vectors as new features.

2.2 Additional Variables

Depending on our preliminary analysis, we found eight other variables that contain information about rating.

- **year** : scaled year variable.
  - From the left plot below, there tends to be more 5-star reviews as time goes by.
- **loc1** : 1 if the restaurant is in the Western United States, otherwise 0.
- **loc2** : 1 if the restaurant is in the Eastern United States, otherwise 0.





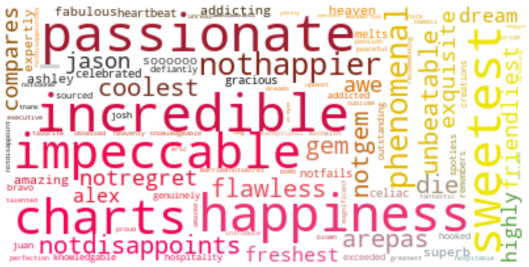
- **Score1 ~ Score5** : Score1[word] = $\frac{\text{P(this word is included in 1-star reviews)}}{\text{P(this word is included in reviews with other stars)}}$, Score2 ~ Score5 are similarly defined.
- **S1 ~ S5**: S1[review] = # of words with high Score1 in the review. S2 ~ S5 are similarly defined on Score2 ~ Score5 respectively.

Word	Variable	1-star	2-star	3-star	4-star	5-star
refund	frequence	115	15	7	4	2
	probability	0.011	0.002	0	0	0
	Score	34.200	1.080	0.300	0.072	0.025
notdisappoints	frequence	0	2	5	43	110
	probability	0	0	0	0.002	0.003
	Score	0	0.116	0.188	0.917	3.870
and	frequence	9196	8691	12851	25604	32071
	probability	0.859	0.886	0.877	0.895	0.886
	Score	0.968	1.000	0.991	1.020	1.000

Intuitively speaking, **"Refund"** is a negative word (you won't ask for a refund if you are satisfied with the restaurant) and **"notdisappoints"** is a positive one while **"and"** contains no information. If we merely consider probability of a word, we will mistakenly think the word **"and"** is important, **"refund"** and **"notdisappoints"** are useless since their probability is close to 0. However, if we use **Score1 ~ Score5** to judge the sentiment of words, **"Refund"** get a high score for 1-star reviews and **"notdisappoints"** for 5-star reviews. **"And"** shows no preference. For each star level, we select 2000 high score words.

We take words with high Score1 value as negative and high Score5 value as positive. The following two word clouds show positive and negative words selected through Score1 and Score5. This method is powerful in splitting positive and negative words.

Positive



Negative

3 Model MSE Comparison

From the table below, LSTM with sentence vectors and additional variables achieves least MSE. This result is based on 100000 observations. When increasing the size of training set, the difference between LSTM and Basic Neural Net increases.

Feature\Model	Linear Regression	Naive Bayes	Basic Neural Net	LSTM	GLM	SVM
vector + ad	0.673	0.974	0.494	**0.493**	0.698	NA
vector	0.720	1.112	0.524	0.526	0.756	0.585
additional	0.836	1.459	0.614	0.612	0.894	NA
frequence	NA	NA	NA	NA	0.864	0.790
tf-idf	NA	NA	0.804	NA	0.836	0.770

Interpretable Model

We use eight additional variables to fit a linear regression model as our interpretable model.

$$y = 3.65 + 0.04 * scale(year) + 0.04 * loc1 + 0.06 * loc2 \\ - 0.11 * S1 - 0.17 * S2 - 0.03 * S3 + 0.03 * S4 + 0.14 * S5$$

From the formula, we can see that year, S4 and S5 have positive effect on the rating while S1, S2 and S3 is negative. This is in line with our intuition.

4 Conclusion

- **Strengths**
  - *Final Model*: Our selection of model and features produces accurate predictions and the inclusion of additional informative variables contributes to the reduction of MSE by 0.033. The final RMSE on Kaggle is around 0.635.
  - *Interpretable Model*: This model is simple and help to find out what makes a review positive or negative.
- **Weaknesses**
  - *Final Model*: We have not experimented much on grid search over various model parameters and leave potential room for further optimizing our results.
  - *Interpretable Model*: Precision is sacrificed for simplicity of this model.
- **Conclusion**
  - We eventually achieved satisfactory prediction accuracy through training LSTM on 100-dimension sentence vector plus carefully selected additional variables. Specifically, features are extracted to capture text order, sentiment and importance. Final combination of model and features is determined based on wide comparison of their performance, i.e. MSE. Our model is likely to further improve with better tuned parameters, such as the size of recurrent neural network, learning rate and so on .

5 Reference

[1]Sida, W. and Christopher D. M.,2012, '*Baselines and Bigrams: Simple, Good Sentiment and Topic Classification*', ACL

[2]Goldberg, Y.,2015, '*A primer on neural network models for natural language processing*', CoRR abs/1510.00726

Contributor

Jianmin Chen:

1. LSTM model
2. sentence vector and additional vector
3. executive summary section 2 and 2.1

Chenlai Shi:

1. SVM, neural network and dimension reduction on word frequency, tf-idf
2. word cloud
3. executive summary section 1.1, 1.2, 3 and 4

Yifan Li:

1. clean the data
2. part of glm regression
3. executive summary section 2.2