## horizontal line



Detection of Fake Spam Reviews on Yelp

05.07.2019

Submitted to: Prof. David Anastasiu

Submitted By -

Kunj Parikh (SJSU ID: 012532065) : kunj.parikh@sjsu.edu

Neha Bindle (SJSU ID: 013763126)

Shrey Patel (SJSU ID: 012430652)

horizontal line

# Abstract

Online reviews play a very important role for decision-making in today's e-commerce. Large parts of the population, i.e. customers read product or store reviews before deciding what to buy or where to buy and whether to buy or not. Because writing fake / fraudulent reviews comes with monetary gain, online review websites there has been a huge increase in tricky opinion spam. Basically, an untruthful review is a fake review or fraudulent review or opinion spam. Positive reviews of a target object can attract more customers and increase sales; negative reviews of a target object can result in lower demand and lower sales. Fake review detection has attracted considerable attention in recent years. Most review sites, however, still do not filter fake reviews publicly. Yelp is an exception that over the past few years has filtered reviews. Yelp's algorithm, however, is a business secret. In this work, by analyzing their filtered reviews, we try to find out what Yelp could do. The results will be useful in their filtering effort for other review hosting sites. Filtering has two main approaches: supervised and unmonitored learning. There are also about two types in terms of the characteristics used: linguistic characteristics and behavioral characteristics. Through supervised learning approach we have tried to make a model which can identify the fake review with almost 70 percent accuracy.

**Introduction**

As the Internet continues to grow in size and importance, the quantity and impact of online reviews is increasing continuously. Reviews can influence people across a wide range of industries, but they are particularly important in e-commerce, where comments and reviews on products and services are often the most convenient, if not the only, way for a buyer to decide whether to buy them. Online reviews can be generated for a variety of reasons. Online retailers and service providers may often ask their customers to provide feedback on their experience with the products or services they have purchased in order to improve and enhance their businesses. Customers may also feel inclined to review a product or service if they had an exceptionally good or bad experience with it. While online reviews can be helpful, blind trust of these reviews is dangerous for both the seller and buyer. Many looks at online reviews before placing any online order; however, the reviews may be poisoned or faked for profit or gain, thus any decision based on online reviews must be made cautiously. Furthermore, business owners might give incentives to whoever writes good reviews about their merchandise or might pay Someone write bad reviews of the products or services of their competitor. These fake reviews are considered spam review and due to the importance of reviews can have a great impact on the online marketplace. Someone to write bad reviews about their competitor’s products or services. As review spam is a pervasive and harmful issue, it is an important but challenging issue to develop methods to help businesses and consumers distinguish true reviews from fake ones.

# System Design & Implementation details

The most challenging part of this project is to extract predictive features from reviews and the corresponding reviewer information. For this particular task we basically worked on the following review centric features -

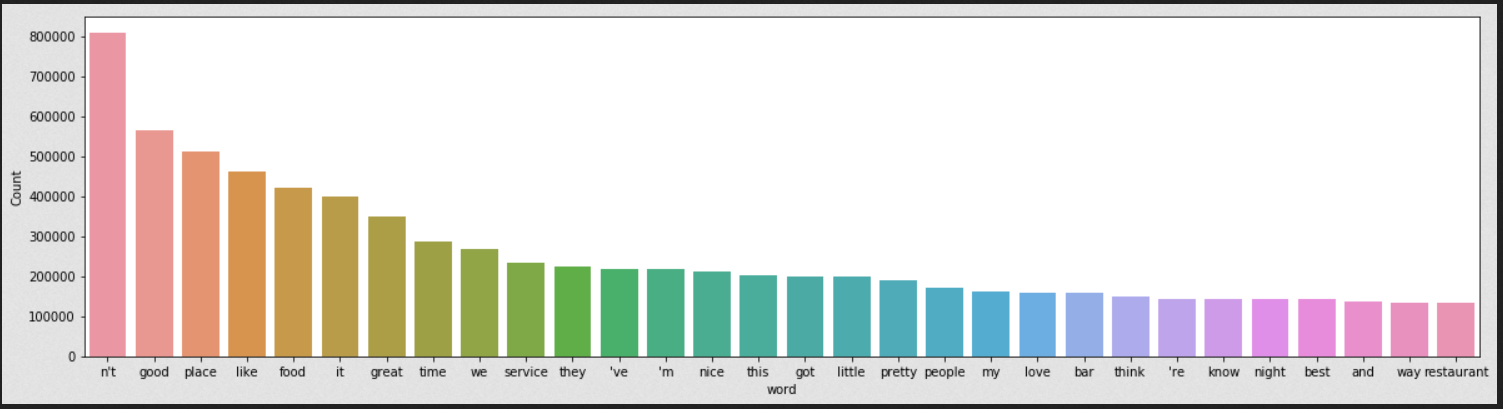
1. **Structural features**: review length, average word length, Standard deviation of the review length.
2. **Semantic features**: We used pretrained model *Textblob* to find out the polarity and sentiment for the review, and also found out the positive and negative sentiment bearing words in the review.
3. **n-gram features**: We have extracted unigram, Bigram, Trigram features from review content.

## Data Preprocessing

* We have majority normal reviews, so we use oversampling to handle biased dataset.
* Use 80-20 train, test split
* Tokenize using Spacy and Keras to split review into words
* Remove punctuations, convert to lowercase.
* Remove stop words using Spacy word-labels (“POS”, “stop word”, etc.)
* Vectorizer to convert word to index, TF-IDF to scale using inverse document frequency.
* Plot word frequency graph and word cloud.

## Feature Engineering for review spam detection

Using the reviews, we add review length feature. Additionally we used a pretrained model from TextBlob to detect sentiment polarity [-1,+1] and sentiment subjectivity [0,1] for each review. Along with this we use other features like rating, usefulCount to train models on features other than review itself.



|  |  |
| --- | --- |
| Normal reviews | Fake reviews |
|  |  |
|  |  |

# Models

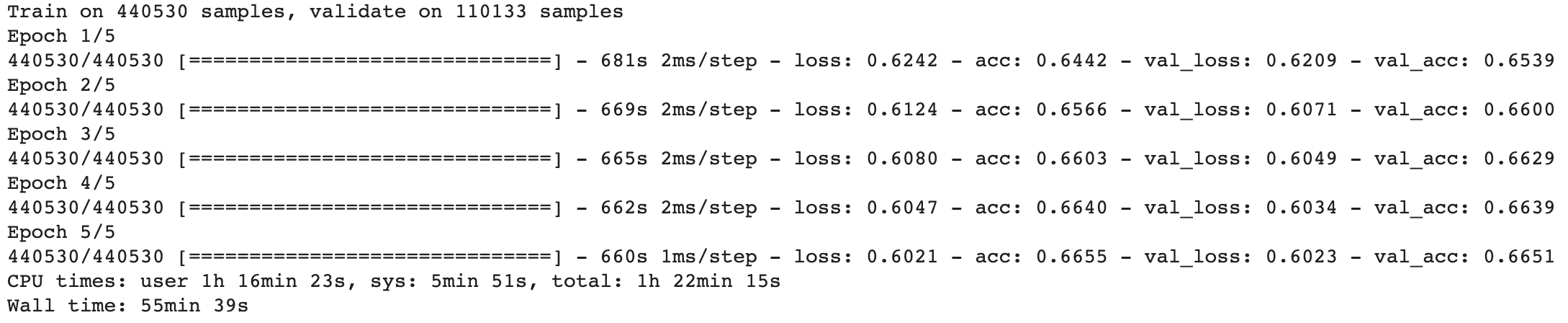
We have tried multiple algorithms and used grid search to find best parameters to get the best model with highest accuracy.

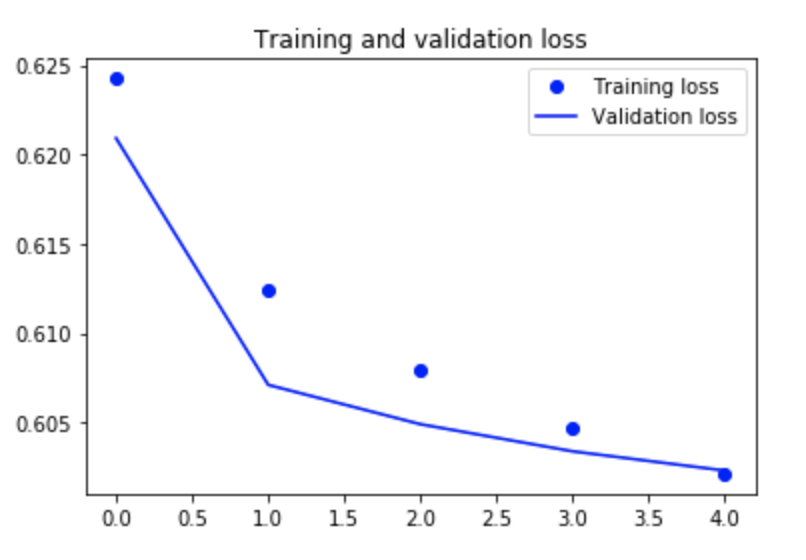
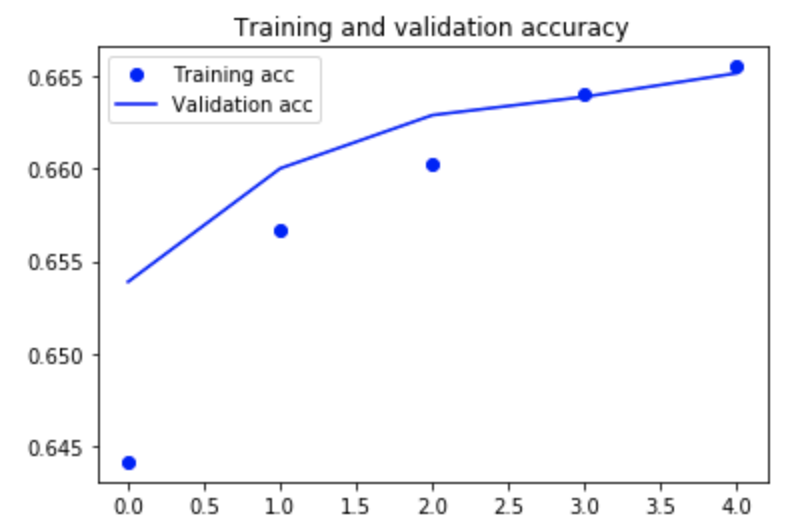
1. **Logistic Regression** - We have used logistic regression on the all the features combined selected through feature selection. It didn’t give good accuracy spo we moved on with other models for better results
2. **Multinomial Naive Bayes** : This is a generative model, in which label is generated using a Bernoulli distribution. For multinomial naive Bayes we initially trained the model on ‘Review Content’ and it gave almost 67 percent accuracy. To improve the accuracy we used GridSearchCV with following parameters to select

*parameters = {'vect\_\_ngram\_range': [(1, 1), (1, 2)], 'tfidf\_\_use\_idf': (True, False), 'clf\_\_alpha': (1e-2, 1e-3)}. We got the best parameter values as ngram\_range':, (1, 2), Tfidf\_\_use\_idf : True, clf\_alpha : .0001.*

After using the above parameters and cross validation accuracy improved to almost 74 percent. We used 3-fold cross-validation in GridSearchCV. We also changed parameters njobs = -1 , so that it can run all jobs in parallel for both fit and predict by using all the available cores to make the execution faster.

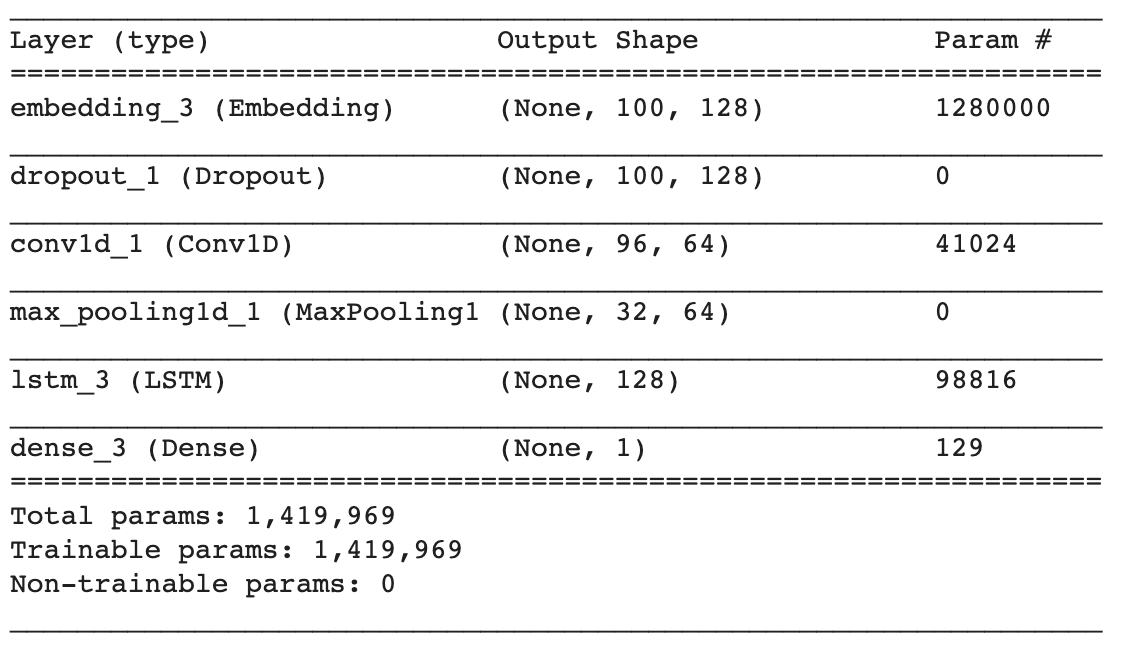
1. **Neural Network with LSTM**: This model had an embedding layer, an LSTM layer, and the LSTM output was fed into a fully connected layer that was hidden. The output size of this layer is 1, which means it will always output 1 or 0, 1 for Spam reviews and 0 for real reviews. 10,000 most common words were passed as features, 64 as the second, and input\_length of 200 as the length of each sequences to the embedding layer. In this first argument came from the second argument from the Embedding layer. We kept 20% chance of dropout. Training model with a batch size of 128, 5 epochs and validation split=0.2, which means that the neural network would learn from 80% of the data and test itself on the remaining 20% of the data.

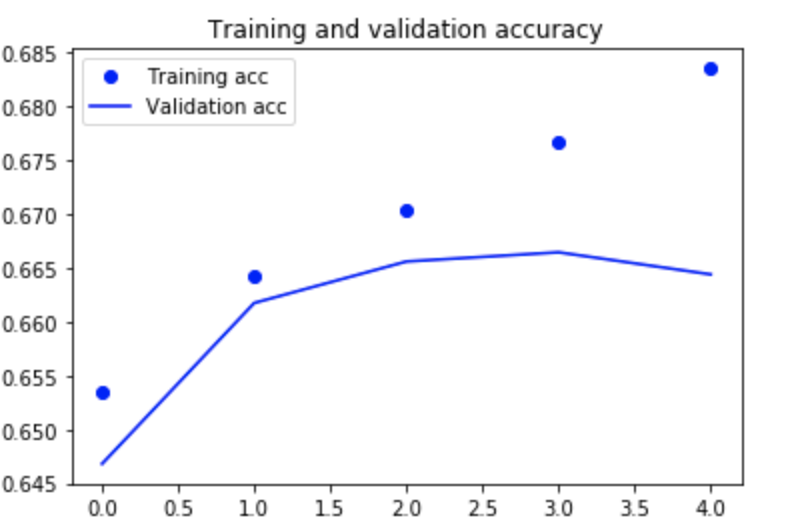
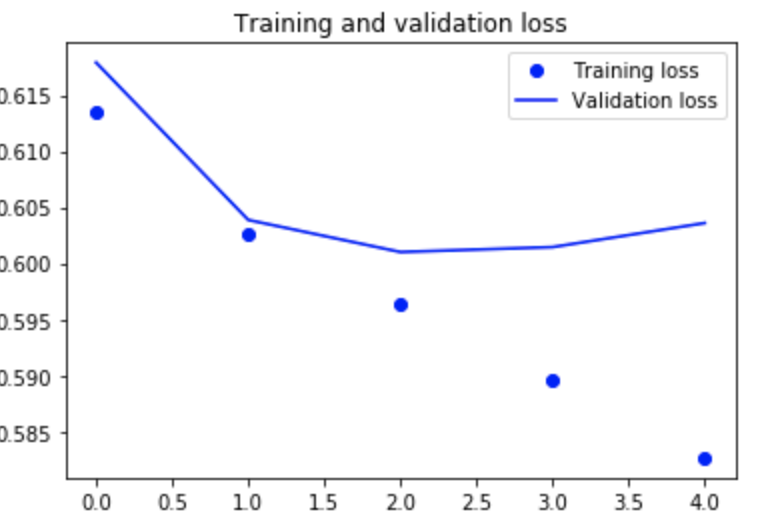


**

For the irst three epochs , the training accuracy was less than validation accuracy which means model was underfitting on the dataset, but in the last 2 epochs, the model generalize well and gave the same training and validation accuracy. This model gave the 66.26 % accuracy on the test set.

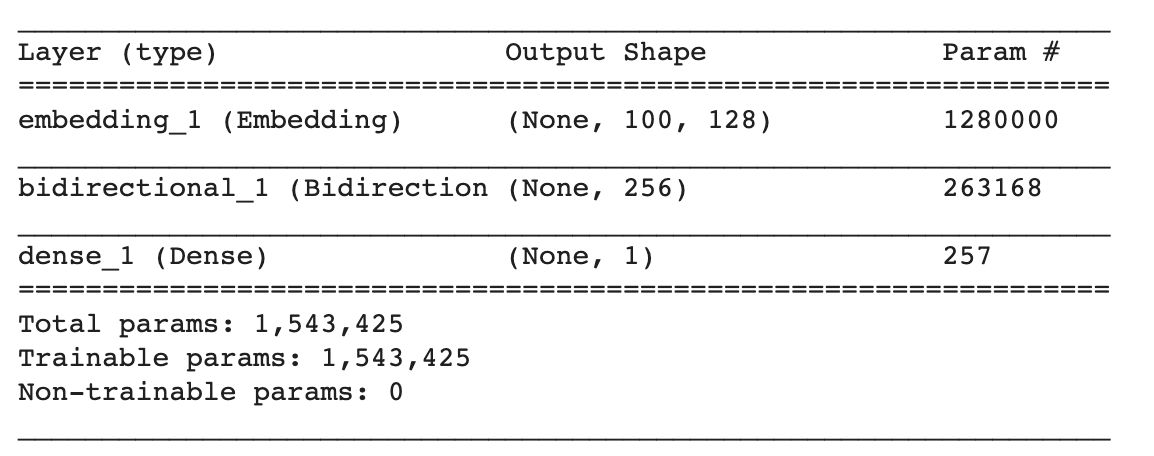
1. **Neural network with an extra 1D CNN on top of LSTM layer:** For this model, concoction of recurrent neural network with LSTM and convolutional neural network. A dropout layer was added just after the embedding layer and added a convolutional layer, a max pooling layer

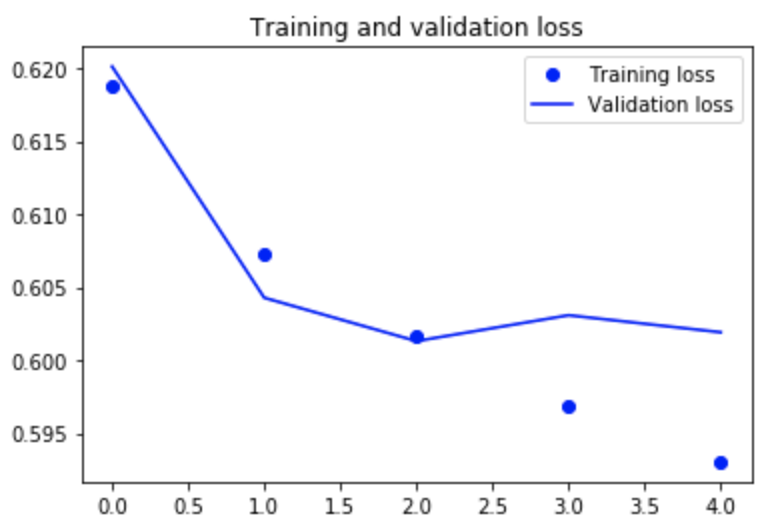
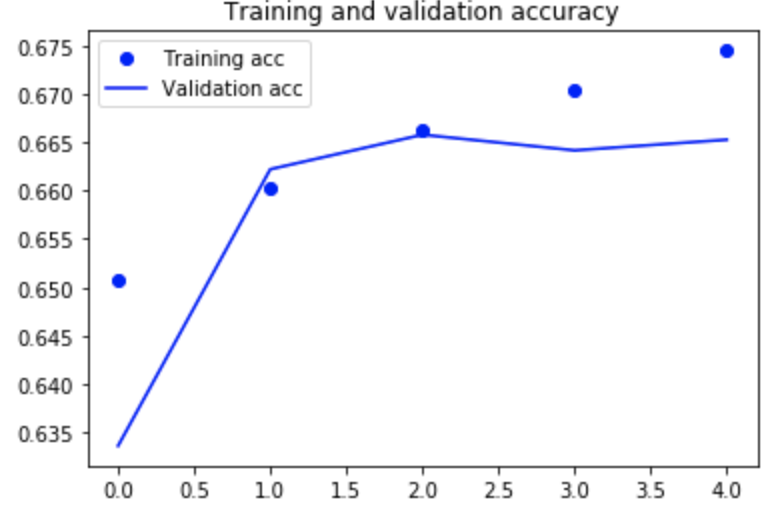




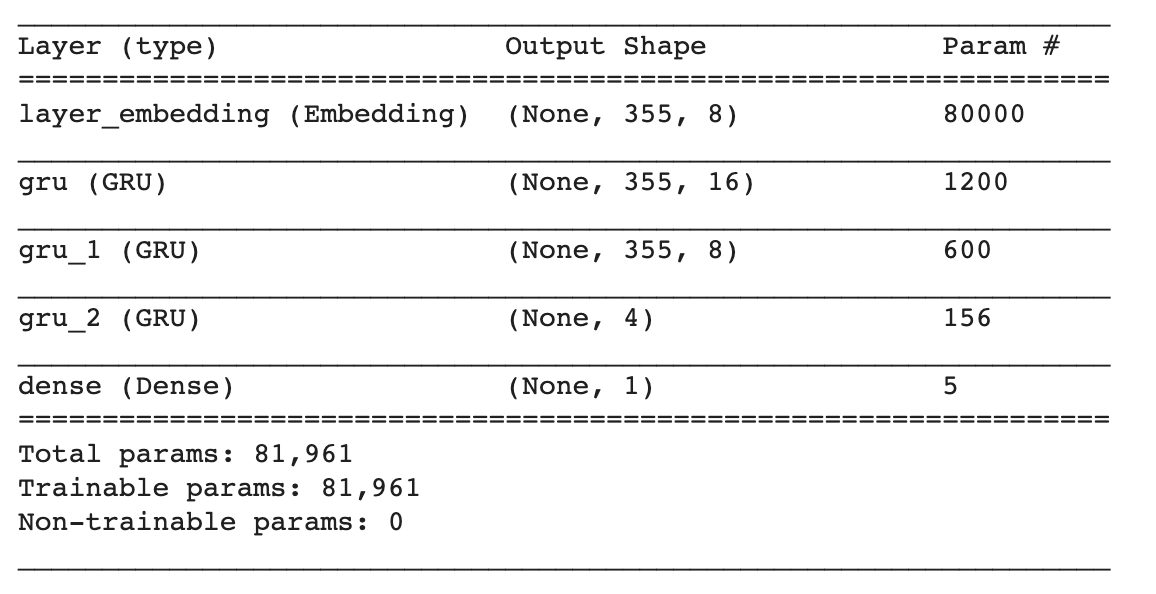
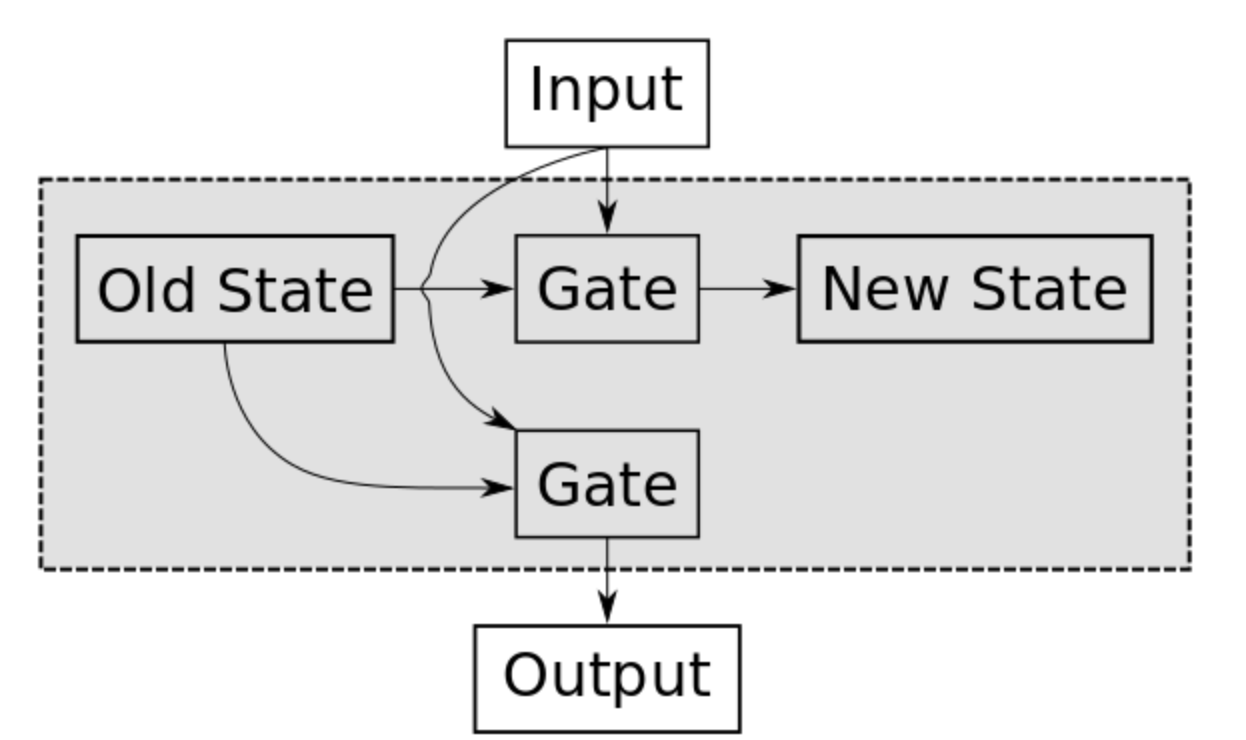
This model gave the similar accuracy of 66.45 percent , and was overfitting after one epoch. However due to speed up was achieved due to additional convolutional layer .

**5) Neural network with using bidirectional recurrent layers:** For this model we used bidirectional recurrent layer as frequently used in natural language processing. The recurrent bidirectional layer potentially provides richer data representations and patterns that may increase accuracy. Bidirectional recurrent layer was added immediately after embedding layer, creating a second separate instance of this layer and using one for chronological processing and another for reversed order processing.





**This model gave the accuracy of 66.45 percent.**

**6)Neural network using Recurrent Neural Network:** We trained this model with the padded sequence and 5 % of training set as validation set. To train the recurring unit, we need to gradually change the gates ' weight matrices so that the recurrent unit provides the desired output for an input sequence. This will be done automatically in Tensorflow.

We got 68 percent accuracy with this model.

7)**XGBoost**: XGBoost is an implementation of gradient boosted decision trees with focus on speed and performance. It uses L2 regularization. We used XGBoost in 2 ways. Once on non-text features like sentiment Polarity, rating, reviewLength etc; this gave us 57% accuracy. Next we tried XGBoost on text data - the CSR IDF matrix, and this gave us better results with 64% accuracy.

8) **Random Forest**: We combined non-text features with CSR matrix by adding them as column features in the matrix. We used this with Random Forest Classifier : 1000 trees, max\_depth of 4. We used bootstrapping in Random Forest instead of using entire dataset. It uses Gini index to split the tree. We got 65.93% accuracy on the combined features dataset.

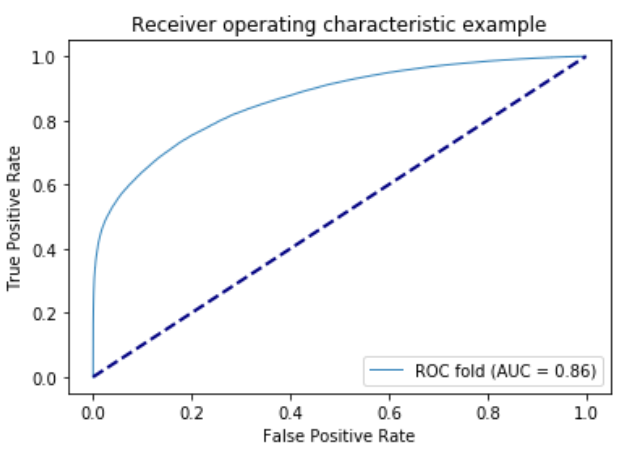
**9)Support Vector Machine**: We tried SVM linear classifier on both type of dataset - review content and non-text features. We used RBF kernel while training on non-text features. With text dataset we got 62.33% accuracy, and with other features like review Length, sentiment Polarity, etc. we got 54.1% accuracy.

**10)SGD Classifier**:

* Objective : To combine the two modelA trained on text data, and modelB trained on other features.
* Model using xa + yb + c, where a and b are predictions of modelA and modelB resp. x and y are the weights to be learnt.
* Split dataset into 64 - 16 - 20 (individual\_train, combine\_train, test)
* Train on 64% data - Use NaiveBayes (A) on review text, XGBoost (B) on non-text features.
* Predict on 16% data - Use the learned modelA and modelB.
* Train on 16% data - Use SGD Classifier and use predictions of modelA and B as input.
* Test using remaining 20% dataset.

# Comparison of Results

|  |  |  |
| --- | --- | --- |
|  | Classifier | F1 Score |
| 1 | Support Vector Machine | 72.00% (Text), 54.1%(other |
| 2 | Stochastic Gradient Descent | 55.56 % |
| 3 | Naive Bayes Classifier | 63.52 % |
| 4 | Naive Bayes Classifier (With Grid search and ngram) | 77.00 % |
| 5 | Random Forest | 65.93 % |
| 6 | Neural Network with LSTM | 66.23 % |
| 7 | Neural Network with 1D CNN on top of LSTM layer | 66.38 % |
| 8 | Neural Network using Bidirectional recurrent layer | 66.45 % |
| 9 | Recurrent Neural Network with GRU | 69.06 |
| 10 | XGBoost | 64% (Text), 57%(Other) |
| 11 | Logistic Regression | 56.27 % |



# Difficult Faced

1. Faced memory leak issue numerous times due to the large dataset.
2. Different versions of libraries and packages, took us a lot of time to make it consistent
3. Combining text and non-text features was a problem. It gave lower accuracy using SGD.
4. Dimensionality reduction on 600k reviews with 200k sparse features using SVD causes out of memory issues. Need to explore other methods to apply SVD on large dataset.

# Conclusion

Contrary to our expectations, adding reviewer-centered features does not significantly boost model performance. We believe this is because most reviewers have only one review record, making most of our reviewer-centered features poorly defined (e.g., maximum number of reviews in a day, standard deviation of ratings). Sometimes using the simple approach works best and that happened with this project, using Grid Search cross validation with Multinomial Naive Bayes Classifier and using the best parameter, gave us the accuracy of almost 77 percent which was the highest amongst all the other classifiers.

# Task Assigned to group members –

1. Kunj Parikh – Data Preprocessing, Model Training and Testing, Report, Presentation, Application Development
2. Neha Bindle – Data Preprocessing, Model Training and Testing, Report, Presentation, Application Development
3. Shrey Patel - Data Preprocessing, Model Training and Testing, Report, Presentation, Application Development

# References

* Arjun Mukherjee, Vivek Venkataraman, Bing Liu, and Natalie Glance. What Yelp Fake Review Filter Might Be Doing. Proceedings of The International AAAI Conference on Weblogs and Social Media (ICWSM-2013), July 8-10, 2013, Boston, USA.
* M. Crawford, T. M. Khoshgoftaar, J. D. Prusa, A. N. Richter, H. A. Najada, "Survey of review spam detection using machine learning techniques", Journal of Big Data, vol. 2, no. 23, Dec 2015.
* <https://scikit-learn.org/stable/supervised_learning.html#supervised-learning>
* <https://github.com/zzhang83/Neural-Network-3-projects/blob/master/Yelp%20Review%20Classification.ipynb>
* <https://github.com/lukas/ml-class/blob/master/keras-cnn/cnn.py>
* <https://towardsdatascience.com/sentiment-analysis-with-text-mining-13dd2b33de27>