# Yelp Fake Review Detection

Anshi Agarwal, Ayushi Bansal, Mridula Gupta

*Department of Computer Science, University of California, Davis*

June 12, 2019

## 1 Introduction

With the increase in dependency of reviews over the internet, there has been a significant decrease in the reliability of those reviews. Businesses have started paying companies/people to generate positive fake reviews/followers and similarly posting negative reviews for the other businesses. Reviews play an important role in making a purchase and these fake reviews can tend to reduce the trustworthiness of the review platforms and the corresponding businesses. This has increased the need for computational tools to provide insights for the reliability of online content. Our aim is to build one such model to counter the problem of identifying fake reviews. We have used the dataset from one of the most popular review site in the field of restaurants.

The previous approaches that have been used for opinion spam detection mainly focused on supervised learning using unigram and bigram features and user behaviour features. Some of the previous works such as [1] have user behaviour features like average review length, standard deviation in ratings, burst features etc. Also, [2] and [3] used classifiers such as SVM and Naive Bayes. Some of the previous works have also been done using supervised classifiers which are graph based models and focus on the social ties underlying the review site. They are in general less effective, but have the advantage that they do not need labelled dataset for training.

In our project we have used word embeddings generated using Doc2Vec and fastText instead of unigram and bigram features along with user behaviour features and other linguistic features. Also, we used a Random Forest classifier for the final classification.

## 2 Approach

**Dataset:** The dataset [4] is collected from Yelp.com for fake review detection. This data includes 608,598 reviews for restaurants from 260,239 reviewers for 5044 products. Reviews include product and user information, timestamp, ratings, and a plaintext review and a label. They have classified the reviews into two classes genuine and fake. The data set used is skewed and has 80k fake reviews and 520k genuine reviews.

**Approach:** For balancing our data, we have used down sampling technique, thus giving 160k reviews in total. And we have cleaned the reviews (text) in our data, like removing stop words, punctuation, empty tokens and removing empty fields. Apart from the reviews we also added review centric features to improve the classification accuracy.

Review Centric Features:

1. Length of review
2. Word count and Sentence count
3. Average of the word length and sentence length
4. Relevant POS count in review
5. Number of capitalized words
6. Number of numeric characters

According to our data set and the required results, we have used 2 word representation learning techniques: **Doc2Vec** and **fastText** to get the corresponding features. Then **Tfidf** (maximum features = 300) is applied on the cleaned review text to generate additional features for Doc2Vec and fastText techniques.

Random Forest classifier: Used to train all the features obtained after above computations in the Doc2Vec and fastText techniques. Using the same classifier helped us compare the results from each technique. We observed that fastText performed the best among all the techniques for our data set with 69.8%.

Addition to the model:

1. Reviewer Centric Features: To further improve our results we also added Reviewer Centric features.
   (a) Average review length
   (b) Number of reviews per day
   (c) Average rating per day
   (d) Standard deviation of ratings
2. Oversampling: In the end we perform oversampling instead of down-sampling the dataset, to improve the accuracy. Over sampling is performed using SMOTE (Synthetic Minority Over-sampling Technique). This refined data set with review centric and reviewer centric features is trained using fastText with Tfidf and RandomForest Classifier, to give the best result.

We have also classified the reviews using Long Short Term Memory [5] recurrent neural network based on the review text only.

**Rationale:** We used oversampling instead of down sampling to avoid losing information on genuine reviews and improve the accuracy of classification. Random down sampling selects random samples of genuine reviews and hence can produce incorrect results.
Also, majority of the supervised classifiers that have been proposed so far for fake reviews detection were using SVMs or Naive Bayes classifier, we used Random Forest classifier because it is more efficient and effective in terms of results, as demonstrated by [6]

# 3 Experiments and Hypothesis

**Hypothesis:** Since our data set is very skewed, we wanted to find out the best way to balance the imbalanced dataset. There are many ways of learning word representations from the reviews. Using our experiments, we wanted to identify which technique is the most meaningful and works best with our dataset. We wanted to analyze the effect of adding reviewer centric features (in addition to review centric) for fake review detection.
Also, we wanted to perform some dataset analysis to infer the differences in the features of fake and genuine reviews, to get a better idea of the mindset of fake reviewers. We also wanted to compare the different machine learning approaches with a neural network model.

**Experimental Design:** To balance our data, we tried down sampling and up sampling our data set and testing our model over both of these data sets to know which technique is better. To find out the best word representation for the reviews, we used Doc2Vec and fastText word embedding techniques. We then compared our classification performance of the learned representation first without any additional features, along-with review features, and then with both review and reviewer centric features. This helped us find the best working model for the Yelp dataset.
To realize the difference between fake and genuine review characteristics, we performed sentiment analysis on the review text and also analyzed POS tags for fake and genuine reviews. We also used LSTM network on the review data to classify them into fake and genuine and compared its results with the above models.

# 4 Results and Analysis

As described in the experimental section, we started with comparing different representation techniques - Doc2Vec and fastText. Using various accuracy metrics, we concluded that fastText learned better representations for the reviews. We also compared our results with the LSTM model and the performance was even worse. Results can be seen in Figure 1.

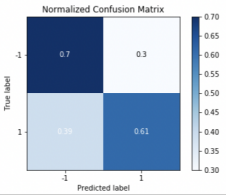
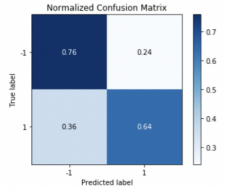
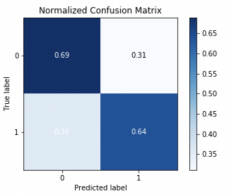
| | Loss | MaP | Score | AuC | Confusion Matrix |
|---|---|---|---|---|---|
| **Doc2Vec (+ Review features) (+Down-sampling)** | 0.315 | 0.607 | 0.655 | 0.71 |  |
| **fastText (+ Review features) (+Down-sampling)** | 0.302 | 0.698 | 0.754 | 0.76 |  |
| **LSTM (+ Review features) (+Down-sampling)** | 0.335 | 0.614 | 0.665 | 0.65 |  |



Figure 1: Comparison of various models for Fake Review Detection

For these experiments, we used review features (like review length, word count, etc.) along with down-sampling the data to make it un-skewed. We also used confusion matrix to compare the accuracy for both fake and non-fake reviews.

We also used various sampling techniques to handle the data and compared our results for the original dataset (7:1 ratio for fake and genuine reviews), down-sampling by removing genuine reviews and over-sampling with additional fake reviews vectors. We compared our results with the review features, and even though the overall accuracy of the original dataset seemed better, looking at the confusion matrix one could clearly see that the accuracy for the fake reviews was really low and was being suppressed by the large number of genuine reviews. Down-sampling performed much better for the fake reviews.

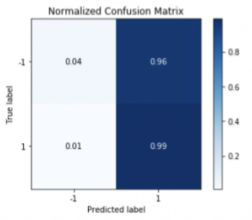
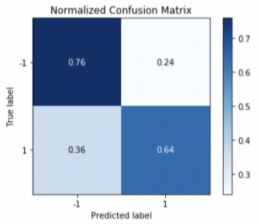
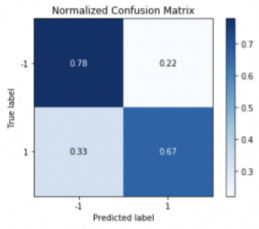
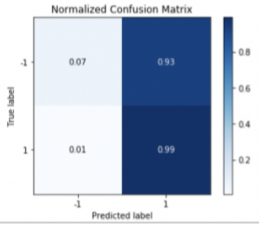
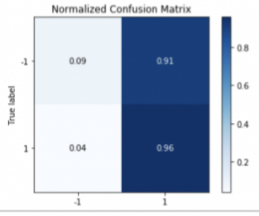
| | Loss | MaP | Score | AuC | Confusion Matrix |
|---|---|---|---|---|---|
| **fastText (+ Review features)** | 0.132 | 0.867 | 0.932 | 0.72 |  |
| **fastText (+ Review features) (+Down-sampling)** | 0.302 | 0.698 | 0.754 | 0.76 |  |
| **fastText (+ Review and Reviewer features) (+Down-sampling)** | **0.277** | **0.790** | **0.722** | **0.80** |  |
| **fastText (+ Review and Reviewer features) (+Over-sampling)** | 0.132 | 0.868 | 0.942 | 0.76 |  |
| **fastText (+Over-sampling)** | 0.155 | 0.845 | 0.901 | 0.61 |  |



Figure 2: Comparison of fastText with various approaches for Fake Review Detection

We also wanted to analyze the importance of reviewer features and the possible improvement over the down-sampled fastText model. While the overall accuracy increased slightly, it did not result in a huge difference. Also as seen in Figure 2, over-sampling the dataset (with both the features and without any features), resulted in low accuracy for the non-fake reviews. Our understanding is that the learned over-samples for the non-fake reviews were not good enough and hence, the model did not perform much better than the original dataset.

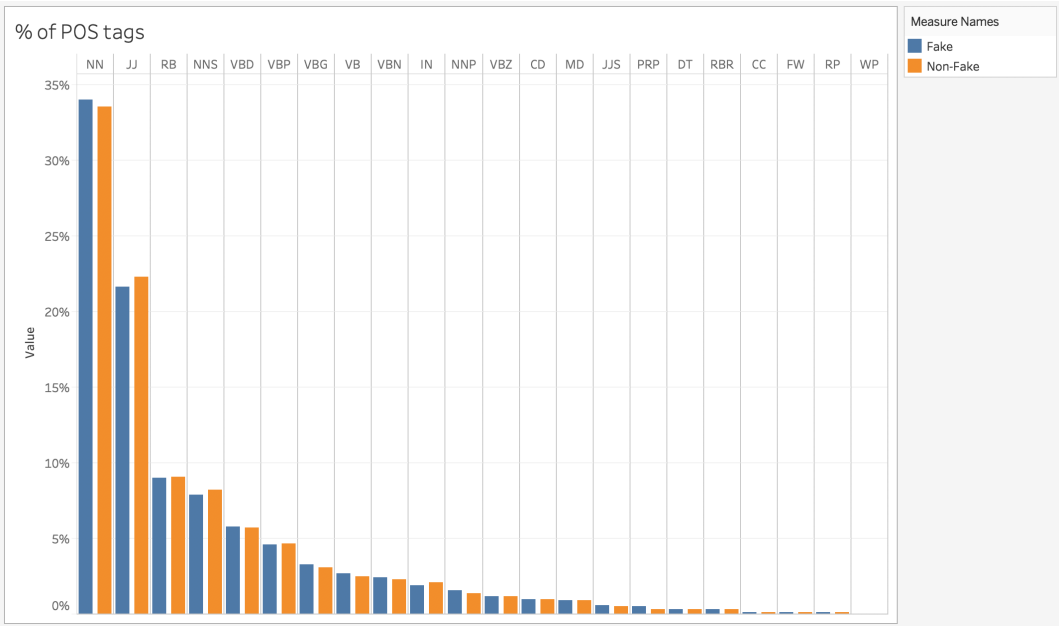



Figure 3: Percentage of POS tags for fake and genuine reviews

In Figure 3, we can see that the dataset is analyzed in terms of the top 22 POS tags present in the dataset for fake (blue colored) and genuine reviews (orange colored). The major inference drawn from the above chart is that the most common tags in the YELP dataset are nouns, adjectives and the percentage decreases drastically from adjectives to adverbs. Also, there is very little difference in the percentages of all POS tags for fake and genuine reviews, so they might not have contributed as a significant feature in differentiating fake from genuine reviews.
With all the experiments that we performed, the best performing model was fastText with both review and reviewer centric features and down-sampling of the fake and genuine reviews.

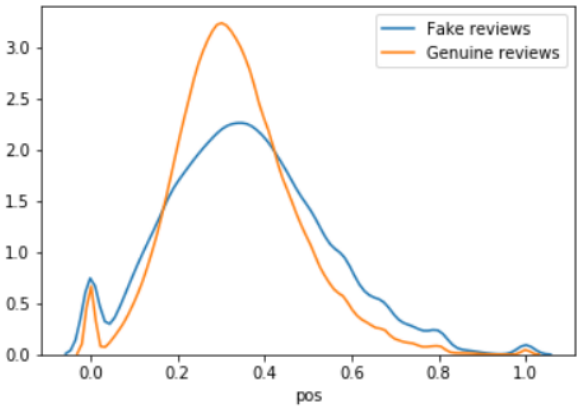



Figure 4: Distribution of reviews with positive sentiments

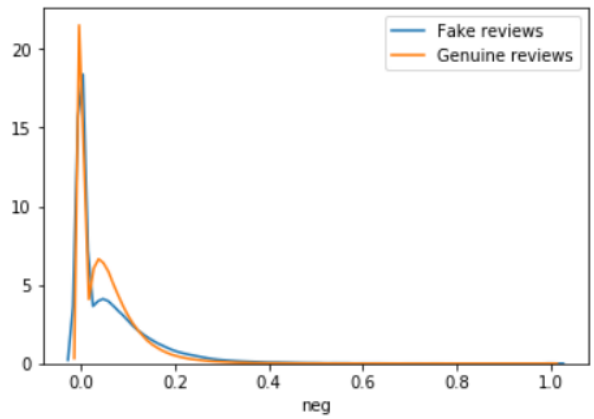



Figure 5: Distribution of reviews with negative sentiments

In Figure 4 and Figure 5, number of reviews are mapped on y-axis and x-axis represents the positivity and negativity score of the reviews. It is quite evident from the graphs that, there are more fake reviews with higher positivity as compared to genuine reviews.

It is quite evident from the experimental results, LSTM didn't perform better. Possible reason might be that in LSTM we are only feeding the review text (as embedding) in the network and ignoring other review centric and reviewer centric features that we generated. Also Figure 4 and Figure 5 help us in concluding that there is not much of a difference in the sentiment of fake and genuine reviews, that is, fake reviewers are very good in mimicking the genuine reviews, so we have to look for other characteristics other than just the review text.

By analyzing our results, we conclude that fastText with tfidf and Random Forest classifier performs the best. Also, down sampling of data for our dataset and problem performs better than over sampling, as the features generated from the review text, when over sampled worsens the results.

## 5 Conclusion

We inferred from our experiments that for balancing the data, down-sampling gave better results as compared to over-sampling in our problem. Intuitively, this might be due to the fact that, over-sampling, which generates random features fails to account for the features specific to review text that characterizes the fake reviewer. After trying out both word embedding techniques mentioned above, fastText proved to be more efficient and learns better representations than Doc2Vec in generating word vectors. LSTM didn't perform well in predicting the fake reviews on our data set, one of the plausible reasons for that being the unavailability of review and reviewer centric features for the network as discussed in the Results section.

The analysis of POS tags and sentiments over the fake and genuine reviews showed that they had low significance in the classification problem. Contrary to our initial assumption, reviewer centric features did not boost the model performance by a significant amount. We infer that the reason behind this was that most reviewers had only single reviews in the Yelp dataset, which makes most of the reviewer centric features we used (such as standard deviation of ratings, maximum number of reviews in a day) irrelevant. We hope these features can be applied to larger dataset in the future for them to improve the performance further. Since

the dataset can be noisily labelled, we can also try out a soft margin classifier like SVM with epsilon in the future.

## References


[1] J. Fontanarava, G. Pasi, and M. Viviani. Feature analysis for fake review detection through supervised classification. In *2017 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, pages 658–666, Oct 2017.

[2] Zehui Wang, Yuzhu Zhang, and Tianpei Qian. Fake review detection on yelp. 2018.

[3] Santosh Kumar Ghosh and Saransh Zargar. Detection of review spam in online review websites. 2015.

[4] Shebuti Rayana and Leman Akoglu. Collective opinion spam detection: Bridging review networks and metadata. In *Proceeding of the 21st ACM SIGKDD international conference on Knowledge discovery and data mining, KDD15*, 2015.

[5] Chih-Chien Wang, Min-Yuh Day, Chien-Chang Chen, and Jia-Wei Liou. Detecting spamming reviews using long short-term memory recurrent neural network framework. pages 16–20, 06 2018.

[6] Julien Fontanarava, Gabriella Pasi, and Marco Viviani. Feature analysis for fake review detection through supervised classification. pages 658–666, 10 2017.