Fake Review Detection

Aleyna BOZACI
Industrial Engineering
150319630
aleynabozacii@gmail.com

Merve YAYIN
Computer Science Engineering
150116051
yayinm8@gmail.com

Mustafa USCA
Mechanical Engineering
150416054
uscamustafa64@gmail.com

Bekir ÖZKAN
150319557
Industrial Engineering
bekirozkan9698@gmail.com

Metehan ERTAN
Computer Science Engineering
150117051
metehan.ertan@hotmail.com

I. Introduction

Online product reviews are a fundamental part of the decision-making process for customers as well as vendors on e- commerce [1]. Prior to purchasing services or goods, customers first review the online comments submitted by previous customers. However, these comments can be deceiving as they can be spam or fake. These misleading reviews can cause huge damages to company reputation and services or goods. This is a strong incentive for people to game the system and manipulate user sentiment by posting fake opinions or reviews to promote or to discredit some target products [2]. Our project will tackle this problem of spam/fake reviews by developing a model, which could classify a given review as either fake or genuine, thereby helping to make more meaningful review information available to the customers.

If customers want to buy a product, they usually read reviews from some customers about the current product. If the reviews are mostly positive, there is a big chance to buy the product. Otherwise, if the reviews are mostly negative, customers tend to buy other products. These fake reviews may have bad impact on company's reputation as there is no relation between the service or product and fake reviews. Also, in modern world fake reviews can be bought. Using bots and giving money to influencer to show their product better companies can create fake reviews for themselves or for their competitor and this results in an unfair competition.

As a result, it is important for companies like Yelp, Amazon, and Airbnb, whose business models strongly depend on accurate reviews, to be able to detect which review are real and which ones are not.

In this paper we propose the methods that we used. We used CNN (Convolutional Neural Network), NB (Naïve Bayes), RF (Random Forest) algorithms. We tried improving these methods in order to achieve more efficiency and better results. At last created an algorithm that combines the results to get a better accuracy.

II. Related Word

III. Approach

We will be using CNN (Convolutional Neural Network), NB (Naïve Bayes), RF (Random Forest) algorithms. We will be improving these methods in order to achieve more efficiency and better results.

IV. Experiment Setup

a. Finding and fixing datasets

We worked on 2 datasets Deceptive Opinion Spam Corpus v1.4 and Yelp dataset. Deceptive Opinion Spam Corpus v1.4 dataset consists of truthful and deceptive hotel reviews of 20 Chicago hotels.

Deceptive Opinion Spam Corpus v1.4 consist of 4 parts:

- 400 truthful positive reviews from TripAdvisor

- 400 deceptive positive reviews from Mechanical Turk

- 400 truthful negative reviews from Expedia, Hotels.com, Orbitz, Priceline, TripAdvisor, and Yelp

- 400 deceptive negative reviews from Mechanical Turk

The Yelp dataset is a subset of Yelp's businesses, reviews, and user data. This dataset is opensource for personal, educational, and academic purposes. Dataset is available as JSON file. This data includes 608,598 reviews for restaurants. This dataset contains reviews from 5,044 restaurants by 260,277 reviewers. In this dataset, there exist 13.22% filtered reviews by 23.91% spammers.

The Deceptive Opinion Spam Corpus v1.4 had enough size, but it was stored in a difficult way. So, we exported every review to its corresponding spamity. So, every truthful and every deceptive was in same directory.

Fixing the Yelp dataset was harder. As the dataset does not only includes the reviews but each user and their followers and what they follow, restaurant information etc. So firstly, we extracted whole review part. But this part was bigger than we needed. So, we cut it so that it is not more than we needed. Each review was stored in an Excel file. So, we extracted them to their corresponding spamity directory by looking at their tags – such as -1 and 1.

b. Run algorithms and record results

For the classification part we decided to use 3 algorithms. All these algorithms are supervised algorithms. Each algorithm is trained and tested for our 2 different datasets. Each algorithm uses each word as a feature as our dataset is a text-based dataset. For each algorithm we splatted data to 0.8 for training and 0.2 for testing.

Before training each review goes into some NLP algorithms. Firstly, we tokenize the review. After that we remove words that are not English. Later, we remove stop words in order to reduce the noise. And lastly, before running algorithms we ran some methods in order to understand our data better. Each method will be shown in results part.

c. Combine algorithms and record results

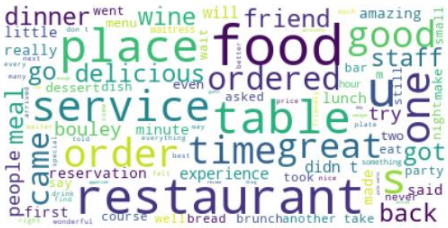
We combined results of these three methods with our algorithm. Firstly, we stored labels and accuracy in a txt file. Later we compared these labels multiplying with their accuracy. Our threshold was 2/3 of total accuracy. If these results are bigger than our threshold than review counts as true. But because we are using 3 different algorithms there is a huge chance that every or most of the algorithms give wrong result. Because of that our algorithm did not gave the accuracy that we wanted. If we had more algorithms, our algorithm would give us better accuracy. Even these results were better than that we expected it was not enough for us to achieve our goal.

V. Experiment Results and Discussion

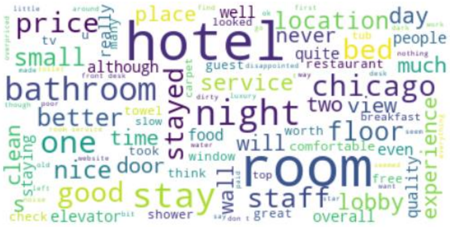
a. Clustering

To find optimal K for k-means clustering we used the Elbow method. After finding the optimal K we created clusters and displayed most used words with WordCloud. As our datasets consist of the same topic most of the clusters got the same most used words. Most used ones change after the third ones.

1. Yelp dataset example



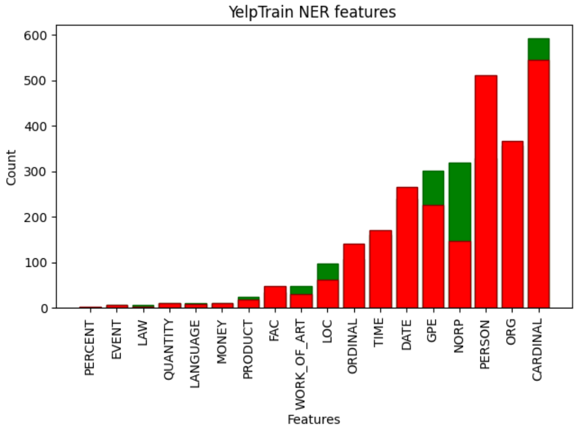
2. Deceptive Opinion Spam Corpus v1.4 example



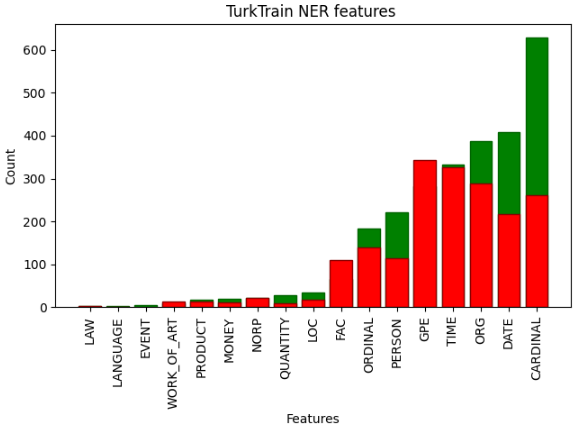
b. NER tagging

As our dataset is a text-based dataset, all data is our feature. Each algorithm uses each word as a feature so we cannot decrease or select any feature. Only methods we can use to decrease feature count are stop word removal and punctuation removal. We can use Named Entity Recognition to determine which attributes are more important and how they affect them.

1. Yelp train dataset



2. Deceptive Opinion Spam Corpus v1.4 train dataset



c. Classification

For the classification part we decided to use 3 algorithms. These algorithms are Multinomial Naive Bayes, Random Forest, and Convolutional Neural Network. All these algorithms are supervised algorithms. We decided that these algorithms will fit great for our project and datasets. Each algorithm is trained and tested for our 2 different datasets. Each algorithm uses each word as a feature as our dataset is a text-based dataset.

For each algorithm we splatted data to 0.8 for training and 0.2 for testing. All Accuracy, Precision, Recall and F1 scores are listed below.

1. Yelp dataset

	Accuracy	Precision	Recall	F1 Score
MNB	80.83	0.808	0.808	0.808
RF	75.62	0.7562	0.7562	0.7562
CNN	73.50	0.7355	0.735	0.7348

2. Deceptive Opinion Spam Corpus v1.4 dataset

	Accuracy	Precision	Recall	F1 Score
MNB	84.17	0.8416	0.8416	0.8416
RF	80.62	0.8062	0.8062	0.8062
CNN	77.62	0.7935	0.7762	0.7729

VI. Conclusion

As you can see from the tables and the data that is above MNB got the best results on all comparisons, RF got the second best and CNN got the worst result. We cannot directly say that MNB is superior to CNN or vice-versa as these statistics can change for each dataset and each method. But for the datasets and methods we can see MNB got the best results in each comparison.

As we use text-based dataset we cannot change the features. This limits our flexibility. MNB is the simplest method that we used therefore it was better on datasets like this. CNN was more complicated and precise therefore it would be better on more featured datasets. We can see this as it was originally designed for image-based datasets. On our dataset CNN overfitted therefore got worse. We saw that on Epoch 10, CNN got 1.0000 accuracy and overfitted the training dataset.

We also had better results on Opinion Spam Corpus (Turk dataset) for all methods. This can be because it had less training data and more testing data. We think that this prevents overfitting.

If we look at the time complexity of these methods. MNB was the fastest followed by RF method. As there are Epochs on CNN it took the most amount of time. Each Epoch nearly took the same amount of time as the MNB method.

To get better results we analyzed the output of these 3 methods. We wanted to combine these 3 methods output in order to achieve better accuracy. But because of our dataset, mostly all our 3 methods give the same wrong output on the same data. We tried to give weight to their outputs to get a higher accuracy. We weighted outputs with their methods corresponding accuracy. If the total of these weighted outputs is greater than a threshold, we counted that review as True. But it did not help because our class attribute is nominal. Multiplying with 1 or 0 eliminated the logic of giving weight. So, it decreased the total accuracy as it mostly eliminated the single correct output when all other 2 are wrong.

VII. References

[1]http://cs229.stanford.edu/proj2019aut/data/assignment_308832_raw/26647583.pdf

[2] https://github.com/sghosh1991/Fake_Review_Detection