**Hotel Review Text Spam Detection**

Robert Wolf

TCSS 455A – Winter 2020

UW Tacoma

Reviews for products are important in the decision-making patterns of consumers. Often differences in reviews and ratings determine a products level of success. This makes websites or applications that garner these reviews targets for Opinion Spam. This seeks to sway public opinion about a product or service by submitting falsified reviews from people who usually have an interest in a competing product.

In this project we will attempt to build a classification model to detect which reviews are spam and which are truthful. The data set we have access contains truthful reviews for hotels from Yelp and Trip Advisor, and deceptive reviews from Amazon’s Mechanical Turk. <https://myleott.com/op-spam.html>

There are 3,200 reviews, half of which are spam and half are truthful. Half of the reviews are positive in disposition, and half negatively review the hotel. The average length of reviews is 806 characters and 148 words, with the longest one being 4,159 characters and 748 words and the shortest being 151 and 25 respectively.

A screenshot of a cell phone

Description automatically generated

Data Cleaning:

To clean the input text, I started with removing punctuation and converting the text to lower case.

Pre-Processing

I have processed the data using NLTK’s stop words tool, removing commonly used words such as “the”, “a”, “an”, “in”, etc. This will reduce the size of the data requiring analysis. I have also stemmed words down by removing morphological affixes from words. An example of this is words such as “staying”, “stayed”, and “stays” would all stem down to “stay.” This again reduces the size of the data while preserving meaning of the words.

Visualizations:

With the data cleaned, I performed several visualizations. Firstly, let’s look at the distribution of word counts and characters in the reviews.

Distribution of Count of Words in Reviews

A screenshot of a cell phone

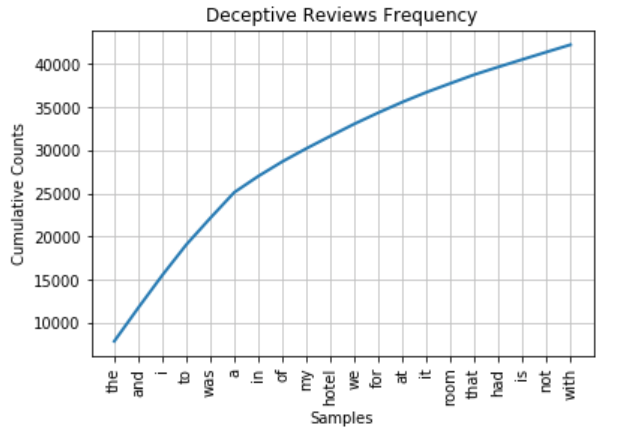
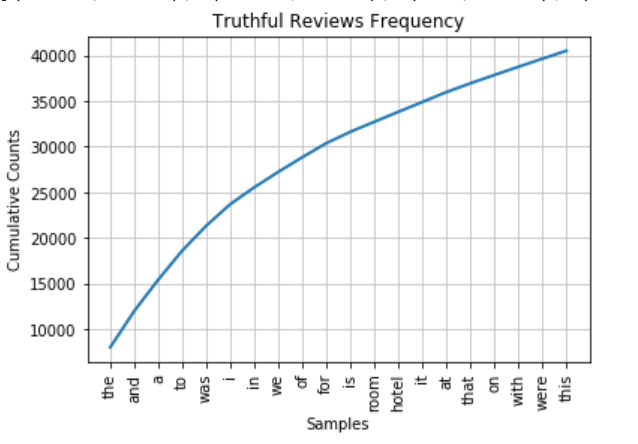
Description automatically generated

Distribution of Count of Characters in Reviews

A screenshot of a cell phone

Description automatically generated

As we can see, these visualizations are not really telling us very much about the difference between deceptive reviews and truthful reviews. Further analysis will be required. Using NLTK, I obtained the 20 most frequently used words from both the original reviews and the cleaned up texts. These do show some interesting differences between the two classes of reviews. Words appear with different frequencies depending on the class, as well as some words not even making the top 20 list for one or the other.



A screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generated

Potential Machine Learning Models:

A Naïve Bayes algorithm might work for this model. It uses n-grams to develop probabilities of words appearing given that the last n words occurred. It makes predictions based on the fact that words appear in certain orders in spam and other orders in truthful texts. This may not be overly accurate, but it would likely be easier to implement and train.

Another possible way to approach this is with word embedding: word embedding transforms words to vector format, and let deep learning recurrent neural networks figure out weights for the vector of the words. This will preserve the state of all previous words when considering the current word. I believe that the Keras library will work well for this, as it is a high-level deep learning library that is able to handle this embedding vectorization and train a model based on that.

Model Training and Evaluation:

To evaluate the performance of one model vs another, I have chosen to use an accuracy measure of how often the model correctly classifies the 20% of the reviews that I will withhold from the data set for validation. Using this measure of effectiveness, I have trained a Select K Best, Naïve Bayes, and Recurrent Convolutional Neural Net using word embeddings. They all had similar performance measures. The most interesting and sophisticated model was the Convolutional Neural Net. It was able to achieve an accuracy of 84.69% when evaluated against the test data after 10 epochs.

A close up of a map

Description automatically generated

However, The best performing of the three was the Naïve Bayes classifier, MultinomialNB from the sklearn package. It achieved an accuracy percentage of 87.81% when validated against the test data. This is good, but not quite as good as some other results that have been achieved in this area. In 2015, Shaswat Rungta was able to achieve a 96.56% accuracy rate by using a bigram + unigram feature set and training a Decision Tree based on that data set (<https://core.ac.uk/download/pdf/80148373.pdf>). I would like to go back and see if I can duplicate the results of this study.

It is worth noting that all 3 of the models train fairly quickly, and all achieved accuracy ratings of above 80%. By further fine-tuning, any of these three would likely be useful in predicting the truthful-ness of the hotel reviews.