

COSC 6397: Project Plan (Spring 2016)

Detection of Reviews Using Semantic Analysis

Avisha Das
P.S. ID: 1403847

March 31, 2016

1 Introduction

In this document I have presented a plan for my Security Analytics project. In this project I will try to apply Security Analytics techniques to a common problem in the consumer market in present times. With the boost of e-commerce and online shopping, opinion and review spamming has been on the constant rise as more and more customers depend their decisions on the online reviews on the commercial websites. These customer feedback and reviews are valuable and of high concern to the manufacturers and retailers as these play an important role in understanding the needs and preferences of the customer.[1]

Opinion spamming is a major issue for commercial websites and retailers as fake reviewers may tend to demean a particular product without having any practical knowledge of the product make and usage while on the other hand, a customer may fall for such fake reviews and end up buying a low quality product. So as a result both the buyer and the seller is prone to such deception and may incur losses due to Review Spamming.

The plan of the project has been divided into the following:

- The Security Problem
- The Dataset Plan
- The Action Plan
- The Literature Survey

1.1 The Security Problem

Review and Opinion Spamming has become one of the major issues in the realm of Cybersecurity with the rise in the popularity of online shopping websites like Amazon and referral websites like TripAdvisor, Yelp, etc. As per common practice, people most often visit review sites while buying a product and almost always opt for the product with maximum number of positive reviews. Therefore, positive reviews can incur in a lot of financial gains for the business and organizations and individuals. This unfortunately is a major attraction for opinion spammers. There are also many fake sites which allow people to write fake reviews. [7]

1.2 The Dataset Plan

The Dataset that I have plan to use for the experiment is a publicly available gold standard dataset[11] of deceptive and truthful reviews of 20 Chicago Hotels. These datasets contain the following:

- 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

For the sentiment analysis, I plan on using the Opinion Lexicon which is a list of positive and negative opinion words or sentiment words for English (around 6800 words)[5].

1.3 The Action Plan

Naive Bayes classifier is a common machine learning classifier technique employed in case of text based classification. Taking a cue from [3,4], the use of a Naive Bayesian classification technique maybe a good starting point for a supervised learning based approach. This assumes that given the review belongs to a particular class, the features extracted from the review are conditionally independent. I plan on applying the Machine Learning techniques on the labeled data set and use the opinion lexicon as a base for my classification. I plan to classify the reviews based on the presence of certain common words expressing sentiment like "good", "great", "clean" etc. expressing positive sentiments and "bad", "worst", "horrible" expressing negative sentiment to classify the sentiment of the review and report the accuracy of classification.

The other machine Learning algorithm that I plan to apply **Logistic Regression** classifier to learn a predictive model on the labeled dataset. As described in [3], logistic regression gives an estimate of the probability that a review is a fake one or not. This can therefore be used to build a model on the labeled dataset available. This method has been chosen by me, because as described in the cited research work, the probability measure can be used to assign a weight to each review and since higher probabilities give a better likelihood of review detection, the ones with higher probabilities can be weighed down to reduce their effects on opinion mining.

1.4 The Literature Survey

I searched the **Google Scholar database** for the list of papers on Sentiment Analysis and Opinon Mining. I have listed my list of references below.

I have mentioned a few related research works in this section and plan to extend my study and survey more in the future. I have added a short description of the techniques and approaches mentioned in the papers.

[1] is a survey of the common and prevalent data mining and machine learning techniques that have been explored and applied to the analysis and detection of Opinion Spamming.

I have also referred to a few online materials: [5,7] are the two online websites that provide a lot of miscellaneous information and list a number of resources and suggested readings about Opinion Mining and Spam Reviews. [5] also gives an insight into the topic of Sentiment Analysis. [11] is the website from where I have extracted my labeled dataset.

The problem of Opinion Spamming is introduced in [2]. The paper identifies two types of Spam Review: *Malicious reviews* which tend to either laud an undeserving product or *Non-reviews* which are very generic advertisements that are not targeted to demean or praise a product or service. The paper uses a large dataset of Review Spams extracted from Amazon.com and uses Logistic Regression classifier to estimate the likelihood that a review belongs to a particular class.

In [6], the author has looked at the aspect of Sentiment Analysis and has discussed the issues that are related to it, thereby calling it a "multi-faceted problem". He has probed deeper to look at the various features that occur in *Opinionated text* and also give a brief description of the technical challenges that make this task more complicated.

Table 1: Top 5 opinion expressing words in Fake Reviews

Top 5 negative words	Top 5 positive words
Ugh	hot
vice	great
heck	commend
din	lean
sin	recommend

In [3], the authors continue on their plan as proposed in [2] and give a much detailed study of the Opinion and Review Spamming, their types and classification. They mention 3 types of common reviews: Untruthful, Brand based and Non-reviews. They also mention methodologies for the detection of duplicates and classification of product based reviews with supervised learning.

In [4], the authors run a Naive Bayesian Classifier on their labeled dataset and then move on to a semi-supervised machine learning approach where they annotate a small section of the review dataset and then employ a Co-training Algorithm that can help classify the unlabeled data and therefore simplify the manual and time consuming task of data annotation.

The book [8] is an almost all comprehensive study that gives an insight into the research and the study that has been done in the field of opinion and review spamming and detection. It covers a wide array of topics on this area.

The authors in [10] study the problem of opinionated text detection with respect to Reviews and opinions. The authors use the dataset collected from Amazon and then use an Unsupervised Bayesian Clustering technique on the unlabeled data. The authors build a Latent Spam model on the data and therefore uses the model in their fake review detection technique.

1.4.1 Classification and Results

In the classification section of the reviews, I had to first preprocess the input data. The following things were the steps that were followed while the dataset was made ready for classification and analysis.

The labeled dataset described in Section 1.2 is actually a collection of 800 fake reviews of hotels, 400 each expressing positive and negative sentiment. Similarly for the case of real reviews. But for processing the classifier much faster on the data I decided to punch the 800 Fake reviews, both positive and negative reviews together and similarly for the 800 Real reviews.

Then I decided to run a frequency calculator based on the Lexicon mentioned in [5]. These two lexicons given here are a list of common sentiment expressing words which occur mostly in Reviews are therefore very necessary in detecting which reviews and what type of reviews contain such words. These words were used in [9]. The main aim of this experiment of mine was to find out which types of reviews contain what kind of common sentiment expressing words.

Below are the tables where I have put down to demonstrate the top 5 commonly occurring words that express opinion of the text. The Table 1 gives the names of the top 5 opinion expressing words based on the dataset of fake reviews chosen in this analysis. Similarly, table 2 gives the top 5 opinion expressing words with respect to the real reviews.

Then for studying the actions of a classifier on the dataset I ran the NaiveBayes Classifier and for this purpose I used the TextBlob package which is a very convenient Python package based on NLTK and used for mainly the purpose of Text classification and Sentiment analysis. For the purpose of studying the classifier's accuracy on the dataset of Fake and Real reviews, I first processed the data available into two comprehensive lists of 800 fake and real reviews each which had a shuffled mix of both positive and negative reviews.

I then partitioned the two datasets into 70% for training and 30% for testing purposes.

Table 2: Top 5 opinion expressing words in Real Reviews

Top 5 negative words	Top 5 positive words
Ugh	hot
vice	great
din	lean
heck	clean
ding	nice

I used a CSV format for a convenient parsing of my input file. Then I ran the NaiveBayes Classifier which takes the frequency of occurrence of the most common words in a document as the features.

I trained and tested the classifier on both the datasets of real and fake reviews. Although it ran without any glitch on the dataset for real reviews. It gave a format error despite running it for 3 times on the test dataset for fake reviews.

I also studied the 5 most informative features that were extracted from the training data set for classification. Below are the snapshots of the results of the classifier on the two data sets. I will try and figure out the issue for the Classifier failure on the Fake review data set.

```

C:\Users\Aishah\Desktop\SPRING 2016\Security Analytics\Project\Scripts>python classification_script.py Fake_train.csv Fake_test.csv
--- 0.0 seconds ---
training...
testing...
Traceback (most recent call last):
  File "classification_script.py", line 28, in <module>
    accuracy_value = cl.accuracy(fp.t)
  File "C:\Users\Aishah\Anaconda\lib\site-packages\txtblob\classifiers.py", line 249, in accuracy
    test_data = self._read_data(test_set)
  File "C:\Users\Aishah\Anaconda\lib\site-packages\txtblob\classifiers.py", line 136, in _read_data
    raise FormatError('Could not automatically detect format for the given '
txtblob.exceptions.FormatError: Could not automatically detect format for the given data source.

C:\Users\Aishah\Desktop\SPRING 2016\Security Analytics\Project\Scripts>python classification_script.py Fake_train.csv Fake_test.csv
--- 0.0 seconds ---
training...
testing...
Traceback (most recent call last):
  File "classification_script.py", line 28, in <module>
    accuracy_value = cl.accuracy(fp.t)
  File "C:\Users\Aishah\Anaconda\lib\site-packages\txtblob\classifiers.py", line 249, in accuracy
    test_data = self._read_data(test_set)
  File "C:\Users\Aishah\Anaconda\lib\site-packages\txtblob\classifiers.py", line 136, in _read_data
    raise FormatError('Could not automatically detect format for the given '
txtblob.exceptions.FormatError: Could not automatically detect format for the given data source.

C:\Users\Aishah\Desktop\SPRING 2016\Security Analytics\Project\Scripts>python classification_script.py Fake_train.csv Fake_test.csv
--- 0.0 seconds ---
training...
testing...
Traceback (most recent call last):
  File "classification_script.py", line 28, in <module>
    accuracy_value = cl.accuracy(fp.t)
  File "C:\Users\Aishah\Anaconda\lib\site-packages\txtblob\classifiers.py", line 249, in accuracy
    test_data = self._read_data(test_set)
  File "C:\Users\Aishah\Anaconda\lib\site-packages\txtblob\classifiers.py", line 136, in _read_data
    raise FormatError('Could not automatically detect format for the given '
txtblob.exceptions.FormatError: Could not automatically detect format for the given data source.

C:\Users\Aishah\Desktop\SPRING 2016\Security Analytics\Project\Scripts>

```

Figure 1: Results of Naive Bayes on Fake Reviews dataset

1.5 Comments and Future study

I will continue working on this topic and find better resources which can reinforce my research. I have added to the Action plan and described my algorithms and how I plan to use them in a little detail in this work. But I have not yet started on my project implementation. I will keep adding to this list as part of my future study.

I plan on the syntactic analysis in the second part of my action plan.

References

- [1] M. Crawford, T. M. Khoshgoftaar, J. D. Prusa, A. N. Richter, and H. A. Najada. Survey of review spam detection using machine learning techniques. In *Journal of Big Data*, 2015.
- [2] N. Jindal and B. Liu. Review spam detection. *WWW*, 2007.
- [3] N. Jindal and B. Liu. Opinion spam and analysis. *WSDM*, 2008.

```

C:\Users\Avisha\Desktop\SPRING 2016\Security Analytics\Project\Scripts>python classificati
l_test.csv
--- 0.0 seconds ---
Training...
Testing...
Writing to file...
0.825
Most Informative Features
    contains(dirty) = True          neg : pos    =    20.1 : 1.0
    contains(rude)  = True          neg : pos    =    18.0 : 1.0
    contains(perfect) = True        pos : neg    =    13.4 : 1.0
    contains(finally) = True        neg : pos    =    13.1 : 1.0
    contains(manager) = True        neg : pos    =    13.1 : 1.0
--- 1783.83999991 seconds ---
C:\Users\Avisha\Desktop\SPRING 2016\Security Analytics\Project\Scripts>

```

Figure 2: Results of Naive Bayes on Real Reviews dataset

- [4] F. Li, M. Huang, Y. Yang, and X. Zhu. Learning to identify review spam. In *IJCAI Proceedings-International Joint Conference on Artificial Intelligence*, volume 22, page 2488, 2011.
- [5] B. Liu. Opinion mining, sentiment analysis, and opinion spam detection. <https://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html>.
- [6] B. Liu. Sentiment analysis: A multi-faceted problem. *IEEE Intelligent Systems*, 2010.
- [7] B. Liu. Opinion spam detection: Detecting fake reviews and reviewers. <https://www.cs.uic.edu/~liub/FBS/fake-reviews.html>, 2012.
- [8] B. Liu. Sentiment analysis and opinion mining. *Synthesis lectures on human language technologies*, 5(1):1–167, 2012.
- [9] B. Liu, M. Hu, and J. Cheng. Opinion observer: analyzing and comparing opinions on the web. In *Proceedings of the 14th international conference on World Wide Web*, pages 342–351. ACM, 2005.
- [10] A. Mukherjee and V. Venkataraman. Opinion spam detection: An unsupervised approach using generative models. Technical report, UH-CS-TR-2014, 2014.
- [11] M. Ott. Deceptive opinion spam corpus v1.4. http://myleott.com/op_spam/.