Detection Of Review Spam in Online Review websites

Santosh Kumar Ghosh
sghosh@cs.stonybrook.edu

Saransh Zargar
szargar@cs.stonybrook.edu

ABSTRACT

A majority of customers rely on the review websites which helps in forming an opinion about the product. Thus, positive or negative reviews have direct influence on the product and this makes online reviews an integral part of the business. This, unfortunately, also gives strong incentives for opinion spamming and thus detection of online review spam becomes important. This paper studies the problem of anomaly detection in review data and approaches the problem using three different methods and provides a comparative study of each technique. It adopts a supervised, semi-supervised and an unsupervised approach towards anomaly detection. It also presents an alternative semi-supervised approach using a co training method for review spam detection which requires only a small amount of positive labeled and unlabeled data for classifier training. Supervised and semi supervised methods require labeled data for classifier training. Labeling data is a tedious task in itself and to avoid this, an unsupervised approach based on Local outlier factor is presented. Finally, this paper analyzes the effect of using different classifiers and feature sets on the performance of supervised and semi-supervised methods and performs a comprehensive evaluation of all the three approaches. The experimental results demonstrate that the proposed, alternative co training method outperforms the other approaches and provides significant improvements.

1. INTRODUCTION

The Web is the greatest repository of digital information and communication platform ever invented. People around the world widely use it to interact with each other as well as to express opinions and feelings on different issues and topics. With the increasing availability of online review sites and blogs, customers rely more than ever on online reviews to make their purchase decisions and businesses to respond promptly to their clients' expectations. Detecting opinion spam is a very challenging problem since opinions expressed in the Web are typically short texts, written by unknown people using different styles and for different purposes. Detecting review spams is a difficult task, particularly because human beings are not always able to reliably determine which reviews are spams. Thus the problem can be stated as: "Given a data set of reviews along with associated reviewers and reviewed products, find reviews that are suspicious and reviewers that are suspicious of spamming activities." Given the incentives of review spamming, the spammers are coming up with more crafty spamming techniques that make spam detection a difficult task.

This project aims at combining the key features from previous works on spam detection and presents an enhanced algorithm for spam detection. The approach we are presenting is a content based approach that takes advantage of the textual content of the reviews. Previous works on content based spam detection have introduced several features that play a key role in identifying spam. The idea is to identify and integrate important features discussed in the earlier approaches and analyze their effect on spam detection.

Employing machine learning methods to detect online review spam usually requires labeled data set for classifier training. These methods are very effective if large amount of labeled instances are available from both classes, deceptive opinions (positive instances) and truthful opinions (negative instances). However, in real scenarios it is very difficult to construct large training data set, as manual labeling is a tedious task and determining the authenticity of the opinions is almost impossible. The human evaluators need to be trained and lots of resources need to be spent for labeling the data. To deal with this restriction, we exploit a semi supervised approach towards review spam detection that utilizes a small amount of manually labeled data to annotate the large set of unlabeled data. It uses a co-training algorithm to train classifiers based on review centric and reviewer centric features. The efficiency of the method relies heavily on how well the classifiers are trained. We observe the impact of using different feature sets, using feature selection, on the classification process and the impact of different classifiers altogether on the spam detection process. Finally, we exploit an unsupervised technique towards spam detection, independent of labeling of reviews. We study the results and provide comprehensive evaluation of each of the techniques.

The rest of the paper is organized as follows: after the introduction of the problem and a brief introduction of the proposed spam detection method, Section 2 presents a review of the relevant work. The data set is discussed in Section 3. In section 4, we discuss the methods used for spam detection along with any background that might be

required for understanding and implementing the techniques. We also discuss our adaptation of the co-training approach. Section 5 presents the experimental results and discusses the strengths and weaknesses of each method. Finally, section 6 outlines the difficulties in implementations, discusses our contributions and provides future work directions.

Keywords

Co-training, Naïve Bayes, Linear SVM, Local outlier factor, feature selection

2. RELEVANT WORK

The method we have proposed is a content based approach meaning it depends on the textual contents of the reviews to identify them as spam or non-spam. We list the relevant work that has been done for spam detection and identification.

2.1 Review Spam Detection

Jindal and Liu[1] conducted the initial study of identifying reviews as spam or non-spam. They presented a categorization of spam reviews and proposed several methods to detect them. They assume that duplicate reviews positive examples of spam review based on which a review spam classifier is learned. They speculated that if a model is learned based on such training data, it would be able to capturing many other variations of type 1 spams (i.e., non-duplicate type 1 spams). The advantage it provided was that it assigned a probabilistic score to each review, which denotes the probability of a review to be spam and it gave better results as compared to the other techniques like SVM, decision tree etc. However, the assumption was too restrictive.

2.2 Review Spammer Detection using rating behaviors

Nguyen, Lim, Liu and Jindal et, al [2] proposed an approach aimed at identifying spammers based on their spamming behavior rather than detecting spam reviews. It adopted a user centric, user behavior driven approach and attempted to model the spamming behavior of reviewers using four main behavior models: Targeting Product, Targeting Product Group, General Rating Deviation and Early Rating Deviation. Each of these behavior-model assigns a score to the reviewer and a cumulative score is obtained. These scores then help in determining whether a reviewer can be considered as a spammer.

2.3 Group Review Spam Detection

Detecting group review spam [3] is based on frequent pattern mining to find candidate groups which behave differently than regular customer groups. The candidate groups are then ranked to identify spammer groups. The proposed method used frequent pattern mining to find candidate groups, then computed spam indicator values based on several features and finally, ranked the discovered candidates using SVM ranking. This method utilized automatic ranking of spammer groups but could not detect subtle differences between spammer a non-spammers.

2.4 Building text classifiers using positive and unlabeled examples

Liu, Dai, li and Yu et al [4] proposed a novel content based approach that addressed the problem of manually labeling large data set for accurate training of classifiers. They proposed a novel approach of building two-class classifiers using only positive and unlabeled examples. Their approach utilized Naive Bayes method and SVM (support vector machine) method. It produced good results outperforming conventional methods for classifier building using a two-step approach.

2.5 Review Spam Detection based on Review Graph

Wang and Sihong et al [5] proposed a novel concept of a heterogeneous review graph to capture the relationships among reviewers, reviews and stores that the reviewers have reviewed. It was different from previous approaches as it did not utilize the review text information. However there were some challenges faced by this method such as, there is no clear distinction between fake and real reviews. It introduced a review graph which consisted of 3 types of nodes; reviewers, reviews and stores. The graph denoted a user's reviews, all the stores he had reviewed and other users' review of that sore. Each node has its own properties like average rating, no. of reviews etc. Although the method could identify subtle spamming activities, the precision was not very large.

3. DATA COLLECTION PROCESS

The data set utilized by our project primarily comes from the Yelp website. It contains data in JSON format and the data which was of particular use to us consisted the following entities: Review Object and User (Reviewer) Object. We have only considered data related to restaurants from yelp dataset ignoring the other ones. For this we parsed out records corresponding to restaurants only. The following steps were involved in the data collection process

1. Filter out all business objects in the business data file having the category “Restaurant” in the records
2. Filter out all the reviews having the filtered business IDs in them

3. Filter out all user IDs from the review data file and use that to fetch details of every user from user data file who had reviewed at least one of the restaurants

After performing the data cleansing we have the data set in the desired structure that can be used in our methods for detecting spam. Some of the initial statistics we observed during data processing are:

1. The approximate number of restaurants is 14,303 which account for about 34% of total businesses

2. The approximate number of reviews that are present in the data set are 7,66,000

3. The approximate number of users who reviewed at least one restaurant is 1,85,000 that is 73% of total users in the data set

We have chopped up the data to be used in the project. We randomly select 100 out of the total 14,303 restaurants available and consider all reviews and reviewers associated with them. Since the data set is very diverse and huge, we plan to begin with a smaller data set initially and then expand to see the effectiveness of our proposed method. For our supervised methods, we need to divide the data set into training set and test data set. We initially began with 5000 reviews and used 4000 reviews as training set and remaining 1000 reviews as test data set. The data set consisted of recommended and non-recommended reviews which we used for training our classifiers for supervised methods.

4. PROPOSED METHOD

In this section we discuss the supervised, semi-supervised and unsupervised approaches that we adopt for review spam detection. Firstly, we introduce the feature set that we used for spam detection. Based on previous works, we selected a set of features that play a key role in determining whether a review is spam or not. This set of features is restricted by the information we could extract from the Yelp website. It can be divided into two categories:

A. REVIEW FEATURES

1. Review Sentiment
2. Keyword Relevance
3. Length of Review Text
4. Subjective vs Objective: Represents how much off topic advertisement or other objective information is contained in the review
5. Entity Count: Denotes the number of products mentioned in the review, can be a good indicator as many times spammers praise one product and degrade/compare another in reviews
6. First person vs second person: fake reviews many times attempt to instruct rather than sharing their own experiences. Ratio of first personal pronouns and second personal pronouns is calculated
7. Similarity Score
8. Product popularity: popularity of a product can also attract spammers. Spam reviews might aim to deflate or inflate a product
9. Rating deviation
10. Text Similarity using bigrams and trigrams

B. REVIEWER FEATURES

1. Rating Deviation: Spammers generally give very low or very high ratings to products. Hence, rating deviations can be used to identify spammers
2. Reviewer Friend count: Spammers tend to work alone with relatively low friend count. As we will observe this assumption is reinforced by the results of our experiments
3. Reviewer Review Count

4.1 SUPERVISED METHOD

We employ two different supervised machine learning methods, namely Naive Bayes and Support Vector Machine. Both these techniques require labeled data to build classifiers that annotate data as spam or non-spam. Manual labeling of data is very resource intensive, requires a lot of time and training and still the authenticity of the labeled data cannot be determined with certainty. This is one of the major drawbacks of the supervised learning methods and to mitigate the effects of these weaknesses we explore a semi supervised approach for spam detection based on a co-training algorithm.

4.2 SEMI SUPERVISED METHOD

We use a semi supervised two view co-training algorithm to annotate the large set of unlabeled data from a small labeled data set. The motivation behind implementing this technique is that co-training algorithm takes advantage of the feature split when learning from labeled and unlabeled data. The feature sets we presented are independent of each other as review features are more focused on content and text of reviews. While reviewer features focus on friend count, rating deviation and review count of each reviewer. Another motivation is that manual labeling of data is labor intensive and resource consuming. Labeling even a small set of data set requires a lot of effort and still we are left with a large set of unlabeled data. Co-training aims at utilizing this small set of labeled data to annotate the unlabeled reviews. Its approach is to incrementally build classifiers over each feature sets. It is a two view algorithm, where the first view is to directly detect

if the review is spam; the other view is to detect if the author of the review is spammer. The major steps of the algorithm are:

1. For each review it uses two views of feature sets. Review features **'F_r'** and reviewer features **'F_u'**. **'L'** is a small set of labelled reviews and **'U'** is the large set of unlabeled reviews.
2. We learn two classifiers, C_r based on review features and C_u based on reviewer features.
3. C_r labels reviews from U based on Fr, **p** positive and **n** negative reviews are from U, $T_{reviews}$.
4. C_u labels reviews from U based on F_u, **p** positive and n negative reviews are from U, $T'_{reviewers}$.
5. Extract reviews $T'_{reviews}$ authored by $T'_{reviewers}$
6. Move $T_{reviews}$ U $T'_{reviews}$ from U to L.

We run the co training algorithm using two different classifiers based on Naive Bayes method and SVM and analyze the results obtained using each method. The results obtained from the co-training method are evaluated based on the evaluation metrics of precision, recall and F-score.

$$\textbf{Precision} = \textbf{Sp} \cap \textbf{Sc}/\textbf{Sp}\ ,$$

$$\textbf{Recall} = \textbf{Sp} \cap \textbf{Sc}/\textbf{Sc}\ ,$$

$$\textbf{F} = 2 * \textbf{Precision} * \textbf{Recall}/(\textbf{Precision} + \textbf{Recall})$$

where, Sc is the set of true review spams, Sp is the set of predicted review spams. The co-training algorithm is simple to implement and the only mathematical background required would be an understanding of the Naive Bayes algorithm and SVM algorithm. As we will observe through the results the co-training algorithm takes advantage of the feature split that is not considered by either Linear SVM and Naive Bayes methods and produces superior results.

4.3 UNSUPERVISED METHOD

Outliers or rare instances are frequently observed in various knowledge discovery and data mining applications. Identifying these outliers has attracted a lot of interest, as it helps in detecting anomalies in various applications pertaining to E-commerce, fraud detection, Stock market analysis, intrusion detection etc. The local outlier factor[] is based on a concept of a local density, where a locality is given by k-nearest neighbors, whose distance is used to estimate the density. By comparing the local density of an object to the local densities of its neighbors, one can identify regions of similar density, and points that have a substantially lower density than their neighbors. These data points are considered to be outliers. The local density is estimated by the typical distance, called the reachability distance, used as an additional measure to produce more stable results within clusters.

Since spammers and review spams are considered as anomalies in the data and LOF provides an effective method for anomaly detection we implemented LOF to study whether the outlier detection problem can be modeled as review spam detection problem. Also, being an unsupervised approach LOF does not require any labeling of data and produces effective results. Since LOF can be applied to a dataset with data points that have same feature set, we cannot run LOF on the entire feature. Thus, we chose to run LOF on the reviewers set and products set separately. The performance of LOF and co-training algorithm produce almost similar results, which also substantiates our hypothesis that in our data set review spammers can also be viewed as outliers.

The motivation behind implementing three different approaches towards review spam detection is to study the strengths and weaknesses of each method and to determine which method is most suitable for spam detection given a data set with specific feature set. We performed feature selection to identify the features that influence spam detection process the most and then added more features incrementally to study the effect each feature has on spam detection. We employ different classifiers on supervised and semi-supervised methods and analyze how they differ based on their classification results. Finally, we implement an unsupervised method to overcome the difficulties faced with supervised and semi supervised methods and compare the performance of each method. In the next section we present the experimental evaluations of the methods discussed above.

5. EVALUATIONS

5.1 SUPERVISED METHOD RESULTS

The results of supervised methods are shown in the graph in **Figure 1**. The machine learning method **Naive Bayes** (NB) performs significantly better as compared with the **Linear SVM**. We used 1000 reviews as training data set for NB and liner SVM method to build classifiers and used these classifiers on a data set of 1000 reviews. We initially selected the four best features by implementing feature selection and observe that the Naive Bayes method clearly outperforms SVM. The four best features suggested by feature selection are Reviewer friend count, Reviewer review count, Bigram measures and Length of the review text. In the subsequent experiments, we increased the feature count and observe that NB method consistently performs better. We evaluate the performance using *Precision*, *Recall* and *F-Score*. **Table 1** shows the result of NB and SVM with different feature sets. We observe little

variation in the performance of NB method as we increase feature sets, which indicates that the other features are dominated by our four best features. The maximum *F-score* by NB method is !!!, while SVM attains a maximum *F-Score* of !!!. When we include only review features and exclude the reviewer features the *F-Score* drops the most in both NB and SVM method.

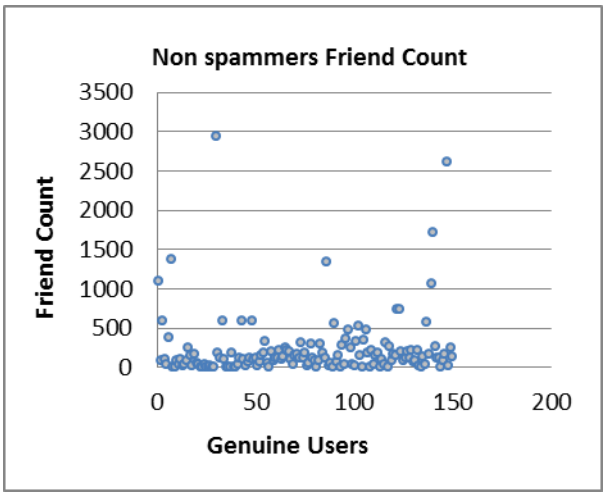

Fig.1 Friend count of genuine users

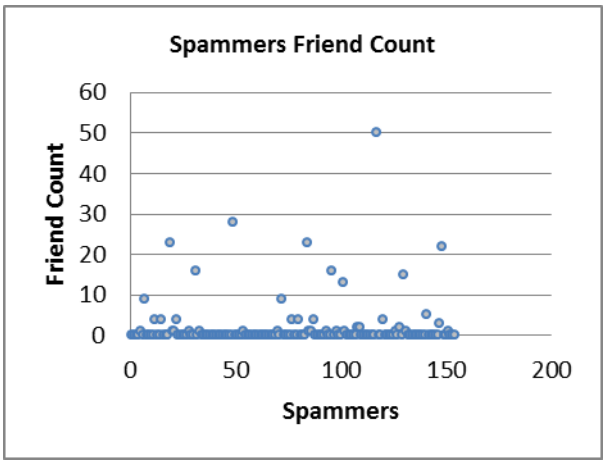

Fig.2. Friend count of spammers

Below we plot the some behavioral features of spammers and non-spammers detected by both NB and SVM method and observe a clear distinction in behavioral pattern that distinguishes spammers with non-spammers. While including the reviewer features like *'Reviewer Friend Count'* and *'Reviewer Review Count'*, we assumed that spammers generally active only during their spamming activities and do not continuously post reviews. Therefore, they have a lesser friend count and review count than genuine users. In contrast genuine users are more active and regular on online forums. This observation is corroborated by our results and demonstrated by the following graphs.

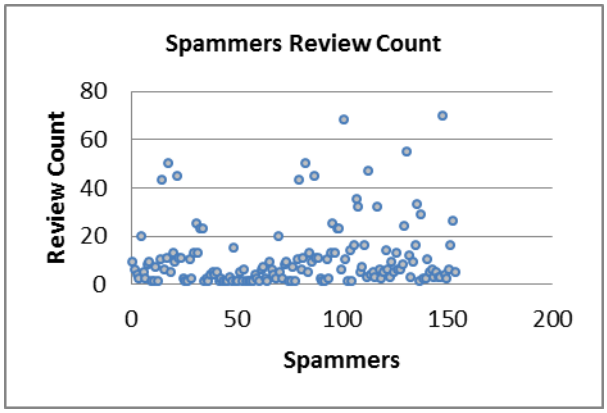

Fig3. Review count of Fake users/Spammers

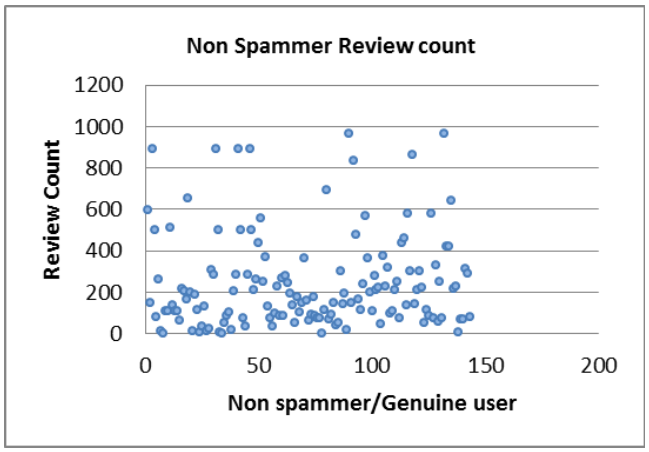

Fig.4 Review count of genuine users

Table 2: Results on Supervised method with feature selection

Number of Features(K)	NB	Linear SVM
K=4	0.585	0.467
K=5	0.557	0.439
K=6	0.534	0.451
K=7	0.542	0.407
K=8	0.538	0.452
K=9	0.546	0.457
K=10	0.551	0.433
K=11	0.558	0.448
K=12	0.561	0.451

5.2 SEMI SUPERVISED METHOD RESULTS

From above section, we have analyzed the effect of various features. In this section, we exploit co-training method to utilize the large number of unlabeled data. We employ both NB and linear SVM method to train two different classifiers and we implement co-training algorithm using both the classifiers. We began with a small set of labeled data of 200 reviews and used this as training set for NB and linear SVM to build classifiers. We consider a data set of 1000 reviews as our test data set. For comparison purposes, this test data set is same as the data set used in supervised learning method. We measure the performance using *Precision, Recall and F-Score*. Table 2 shows that co-training method using NB classifier performs the best as compared to other methods we have discussed.

The reason co-training performs better than the supervised learning methods is because it exploits the independent division of the feature set as mentioned in []. Our feature set is divided into two separate classes, review features and reviewer features and each set of features is capable of classifying the data set on its own. However, the supervised approaches subsume this observation and do not exploit the independent nature of the feature sets.

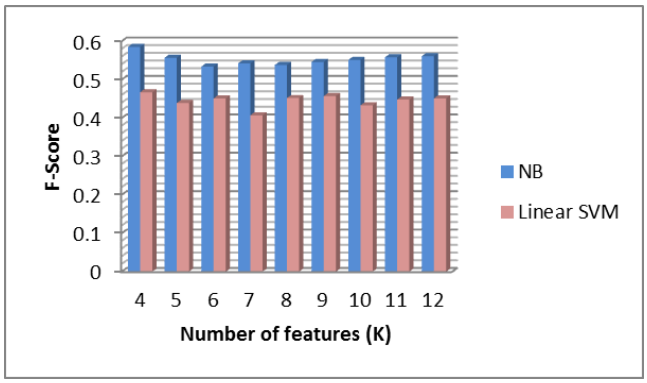

Fig 5. F-score of NB and Linear SVM for different feature sets
.

Table 2: Results on semi supervised method

	Precision	Recall	F-Score
NB	0.467	0.705	0.561
SVM	0.335	0.689	0.451
Co-Training(NB)	0.453	0.802	0.579
Co-Training(linear SVM)	0.413	0.692	0.518

5.3 UNSUPERVISED METHOD RESULT

We now model the spam detection problem as outlier detection problem and try to identify the correlation between the two. Since, spams can be considered as anomalies we assumed that LOF method should be able to identify spams as outliers. Also, being an unsupervised technique we do not require labeled data for spam detection. We pick 500 reviews from our test data set of 1000 reviews and perform LOF on the test data set. Since LOF can be applied to a dataset with data points that have same feature set. We chose to run LOF on the reviewers set and review set separately. We observe that with very few exceptions, the non-spam data points have values close to 1 and spam data points have prominent higher values. These

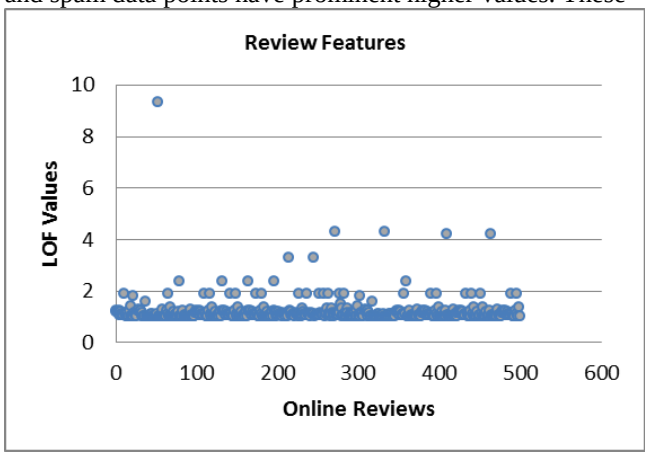

Fig 6. LOF values of 500 online reviews based on review features

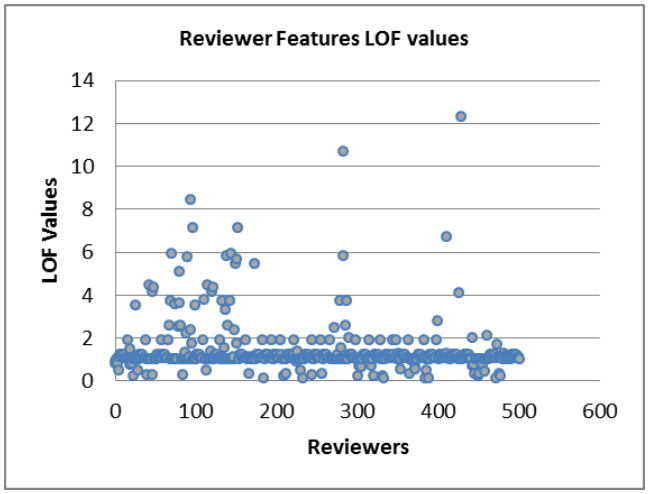

Fig 7. LOF values for 500 randomly selected reviewers based on reviewer features

values correspond with the results of co-training algorithm. The LOF values are illustrated in the below graph.

6. CONCLUSIONS

In this project we tackled the problem of spam detection using three different approaches and evaluated the performance of each approach on our data set. We first

defined a feature set that we used to identify spams. Then we performed feature selection to identify the most important features that help in distinguishing spams from non-spams. The feature selection process and the experimental results corroborated our assumptions regarding the significance of 'Reviewer features'. However, we also observed that some of the features we introduced, like 'entity count', do not have much impact on the spam detection process. We also studied the performance of different classifiers for spam detection using the same feature set. The results demonstrated that NB based classifiers performs better than liner SVM and thus the choice of classifier is also of significance to spam detection. But the problem of manually labeling reviews to train classifiers still persisted.

We observed that the feature set we selected can be divided into two classes which can individually be used to classify out data set. So, we adopted a semi supervised co-training approach which takes advantage of the same and is also easy to implement and requires considerably less amount of labeled data. We used NB based and Linear SVM based classifiers in the co-training algorithm and obtained better results than supervised methods. Furthermore, we applied a well-known outlier detection algorithm (LOF) to the problem whose results turn out to be consistent with the output of co-training method.

There are a few interesting directions to pursue as future work. First, we can enrich our dataset by collecting information about any given product (e.g., hotels) from multiple review websites, and observe if the results of our experiment hold for different data sets. Secondly, we can perform supervised and semi supervised methods on different data sets to identify whether in the presence of independent feature set like ours, co-training always performs better. Thirdly, we can enhance the feature set and study the impact of those features on spam detection. Furthermore, we can implement other classification techniques with co-training apart from NB and SVM and study how the performance varies. Finally, instead of a two view co-training algorithm we can present more views based on more independent feature sets and analyze the performance.

REFERENCES

[1] Nitin Jindal and Bing Liu Department of Computer Science University of Illinois at Chicago nitin.jindal@gmail.com, liub@cs.uic.edu. Analyzing and Detecting Review Spam.

[2] Ee-Peng Lim eplim@smu.edu.sg, Viet-An Nguyen vanguyen@smu.edu.sg, Nitin Jindal nitin.jindal@gmail.com, Bing Liu liub@cs.uic.edu, Hady W. Lauw hwlauw@i2r.a-star.edu.sg. Detecting Product Review Spammers using Rating Behaviors.

[3] Arjun Mukherjee, arjun4787@gmail.com, Bing Liu liub@cs.uic.edu , Junhui Wang jwang@math.uic.edu, Natalie Glance nglance@google.com, Nitin Jindal nitin.jindal@gmail.com, Detecting Group Review Spam

[4] Fangtao Li {fangtao06, yangyiycc}@gmail.com, Minlie Huang {aihuang, zxy-dcs}@tsinghua.edu.cn, Yi Yang and Xiaoyan Zhu

[5] Guan Wang gwang26@uic.edu, Sihong Xie sxie6@uic.edu, Bing Liu liub@uic.edu, Philip S. Yu psyu@uic.edu, Review Graph based Online Store Review Spammer Detection

[6] Michael Luca mluca@hbs.edu, Georgios Zervas zg@bu.ed

[7] DonatoHern´andezFusilier, RafaelGuzm´anCabrera Divisi´on de Ingenier´ıas Campus Irapuato-Salamanca. Universidad de Guanajuato Mexico. {donato,guzmanc}@ugto.mx, ManuelMontes-y-G´omez Laboratorio de Tecnolog´ıas del Lenguaje. Instituto Nacional de Astrof´ısica, ´Optica y Electr´onica. Mexico. mmontesg@inaoep.mx, PaoloRosso Natural Language Engineering Lab., ELiRF. 2Universitat Polit`ecnica de Val`encia Spain. prosso@dsic.upv.es. UsingPU-LearningtoDetectDeceptiveOpinionSpam

[8] Kamal Nigam School of Computer Science Carnegie Mellon University Pittsburgh, PA 15213 knigam@cs.cmu.edu, Rayid Ghani School of Computer Science Carnegie Mellon University Pittsburgh, PA 15213 rayid@cs.cmu.edu. Understanding the Behavior of Co-training