Improved Techniques for Online Review Spam Detection

Smriti Singh

Department of Computer Science and Engineering
National Institute of Technology Rourkela
Rourkela-769 008, Odisha, India.

Improved Techniques for Online Review Spam Detection

Thesis submitted in partial fulfillment
of the requirements for the degree of

Master of Technology

Under the Dual Degree Programme

in

Computer Science and Engineering

by

Smriti Singh

(Roll: 710CS1033)

under the guidance of

Prof. Sanjay Kumar Jena

Department of Computer Science and Engineering
National Institute of Technology Rourkela
Rourkela-769 008, Odisha, India.

May‘ 2015

Declaration by the Student

I certify that:

- The work enclosed in this thesis has been done by me under the supervision of my project guide.
- The work has not been submitted to any other Institute for any degree or diploma.
- I have confirmed to the norms and guidelines given in the Ethical Code of Conduct of National Institute of Technology, Rourkela.
- Whenever I have adopted materials (data, theoretical analysis, figures or text) from other authors, I have given them due credit through citation and by giving their details in the references.

Name: Smriti Singh

Date:

Signature

Department of Computer Science and Engineering
National Institute of Technology Rourkela
Rourkela-769 008, Orissa, India.

May 5, 2015

Certificate

This is to certify that the work in the thesis entitled ***Improved Techniques for Online Review Spam Detection*** by ***Smriti Singh*** is a record of an original research work carried out under my supervision and guidance in partial fulfilment of the requirements for the award of the degree of **Master of Technology, under the Dual Degree Programme**, in Computer Science and Engineering. Neither this thesis nor any part of it has been submitted for any degree or academic award elsewhere.

Sanjay Kumar Jena
Professor
Department of CSE, NIT Rourkela

Acknowledgment

I owe deep gratitude to the ones who have contributed greatly in completion of this thesis.

Foremost, I would also like to express my gratitude towards my project advisor, Prof. Sanjay Kumar Jena, whose mentor-ship has been paramount, not only in carrying out the research for this thesis, but also in developing long-term goals for my career. His guidance has been unique and delightful. I would also like to thank my mentor, Jitendra Rout Sir, who provided his able guidance whenever I needed it. He inspired me to be an independent thinker, and to choose and work with independence.

I would also like to extend special thanks to my project review panel for their time and attention to detail. The constructive feedback received has been keenly instrumental in improvising my work further.

I would like to specially thank my friend Shaswat Rungta for his profound insight and for guiding me to improve the final product, as well as my other friends for their support and encouragement.

My parents receive my deepest love for being the strength in me.

Smriti Singh

Abstract

The rapid upsurge in the number of e-commerce websites, has made the internet, an extensive source of product reviews. Since there is no scrutiny regarding the quality of the review written, anyone can basically write anything which conclusively leads to Review Spams. There has been an advance in the number of Deceptive Review Spams - fictitious reviews that have been deliberately fabricated to seem genuine. In this work, we have delved into both supervised as well as unsupervised methodologies to identify Review Spams. Improved techniques have been proposed to assemble the most effective feature set for model building. Sentiment Analysis and its results have also been integrated into the spam review detection. Some well known classifiers have been used on the tagged dataset in order to get the best performance. We have also used clustering approach on an unlabelled Amazon reviews dataset. From our results, we compute the most decisive and crucial attributes which lead us to the detection of spam and spammers. We also suggest various practices that could be incorporated by websites in order to detect Review Spams.

Keywords: Review Spam, Spam Detection, Opinion Spam, Sentiment Analysis

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 What is Review Spam?

Online product reviews have become an indispensable resource for users for their decision making while making online purchases. Product reviews provide information that impacts purchasing decisions to consumers, retailers, and manufacturers. Consumers make use of the reviews for not just a word of mouth information about any product, regarding product durability, quality, utility, etc. but also to give their own input regarding their experience to others. The rise in the number of E-commerce sites has lead to an increase in resources for gathering reviews of consumers about their product experiences. As anyone can write anything and get away with it, an increase in the number of Review Spams has been witnessed. There has been a growth in deceptive Review Spams - spurious reviews that have been fabricated to seem original [1] . These reviews produced by people who do not have personal experience on the subjects of the reviews are called spam, fake, deceptive or shill reviews. These spammers publish fictitious reviews in order to promote or demote a targeted product or a brand, convincing users whether to buy from a particular brand/store or not[2].

In the last few years, Review Spam Detection has gathered a lot of attention. Over the past few years, consumer review sites like Yelp.com have been removing spurious reviews from their website using their own algorithms. Both supervised as well as unsupervised learning approaches have been used previously for filtering

of Review Spams. For the purpose of training the features for machine learning approaches, linguistic and behavioural features have been used.

There are two distinct types of deceptive review spams:

1. Hyper spam, in which fictitious positive reviews are rewarded to products to promote them
2. Defaming spam, where unreasonable negative reviews are given to the competing products to harm their reputations among the consumers [3]

Specifically, the reviews that have been written either to popularize or benefit a brand or a product , therefore expressing a positive sentiment for a product, are called positive deceptive review spams. As opposed to that, reviews that intend to malign or defame a competing product expressing a negative sentiment towards the product, are called negative deceptive review spams[4].

1.2 Challenges in Review Spam Detection

Table 1.1: Challenges in Review Spam Detection

Traditional Cues	Shortcomings
Review features (bag of words, ratings, brand names reference)	Hard for human, not to mention machines
Reviewer features (rating behaviors)	Poor if one wrote only one review
Product/Store features	Tell little about individual reviews
Review/reviewer/store reinforcements	Fails on large number of spam reviews with consistent ratings
Group spamming	No applicable on singleton reviews
Singleton reviews detection	Finds suspicious hotels, cant find individual singleton spam

1.3 Motivation and Objective

Individuals and organizations increasingly use reviews from the social media for:

1. For making decisions relating to product purchases

2. For product designing and marketing
3. To make election choices
4. 31% of consumers read online reviews before actually making a purchase (rising)
5. By the end of 2014, 15% of all social media reviews will consist of company paid fake reviews

Figure 1.1: Review Websites

The reviews that have been positively written, often bring lot of profits and reputation for the individuals and the businesses. Sadly, this also provides an incentives for the spammers to be able to post fake or fabricated reviews and opinions. Unwarranted positive reviews and unjustified negative reviews, is how opinion spamming has become a business in recent years. Surprisingly there are a large number of consumers who are completely wary of such biased, paid or fake reviews.

Figure 1.2 shows an advertisement by Belkin International, Inc which published an advertisement for writing fictitious reviews on the amazon.com website. (65 cents/review) on Jan 2009.

The effectiveness of opinion mining relies on the availability of credible opinion for sentiment analysis. Often, there is a need to filter out deceptive opinion from the spammer, therefore several studies are done to detect spam reviews. It is also problematic to test the validity of spam detection techniques due to lack of available annotated dataset. Based on the existing studies, researchers perform two different

Figure 1.2: An example practice of review spam

approaches to overcome the mentioned problem, which are to hire annotators to manually label reviews or to use crowdsourcing websites such as Amazon Mechanical Turk to make artificial dataset. The data collected using the latter method could not be generalized for real world problems. Furthermore, the former method of detecting fake reviews manually is a difficult task and there is a high chance of misclassification.

Our main objectives are:

1. To investigate some of the most novel techniques for Spam Detection in online reviews.

2. Our main objective is to build the most effective features set for training model.

3. To detect Review Spams using well known classifiers for labelled dataset.

4. Incorporating aspect-based opinion mining and Sentiment Analysis techniques into our Review Spam Detection methods.

5. Also device an unsupervised method of Review Spam Detection using clustering on unlabelled dataset.

1.4 Problem Statement

Our main goal is to devise automated methods to detect review spams in product websites using review text based as well as reviewer based methods. We obtain the most apt datasets required for the study of the same. We try to obtain the feature sets that can best represent and distinguish the spams from ham(non-spam reviews). We then follow both supervised and unsupervised methodology to obtain review spams from the dataset. We also include sentiment analysis methodology into our review spam detection. Lastly, we compare our analysis obtained from taking various types of feature sets based on review text, sentiment scores, reviewer features, as well as the combined method.

1.5 Thesis Organisation

The present thesis is organised into eight chapters. ***Chapter 1*** presents introduction to review spam and the challenges the occur during review spam detection. ***Chapter 2*** presents a Literature review on review spam, types of review spams and spammers. It also includes a review of the related done done in this field. ***Chapter 3*** highlights the automated approaches to deceptive review spam detection. It explains several methodologies used in supervised review spam detection such as POS tags, text method, etc. In ***Chapter 4*** new features have been proposed that can successfully classify our dataset. It also describes the dataset that has been used and its collection method. It also explains the classifiers used for the classification. ***Chapter 5*** displays the results obtained using different feature sets for the same dataset and compares the same. ***Chapter 6*** proposes a modified k-NN clustering approach that is applied to a new dataset collected from Amazon, having reviewer information as well. This chapter also analyses the new dataset and makes several useful observations. It also displays the result obtained from the unsupervised learning method used on the Amazon review spam dataset. ***Chapter 7*** concludes the work done, highlighting the contributions and suggests directions for possible future work on review spam detection.

Chapter 2

Literature Review

2.1 Types of Spams

2.1.1 Email Spam

Direct mail messages are used to target individual users in Email Spam. The list for email spams is often prepared by scanning the web for Usenet postings, web search of addresses as well as stealing of web addresses.

2.1.2 Comment Spam

Another category includes, comment spam which is widely used by spammer by posting comments for their nefarious purpose.

2.1.3 Instant Messaging Spam

This type of spam makes use of instant messaging systems. Instant messaging is a for of chat based direct communication between two people in real time, using either personal computers or any other devices. The network communicates messages only in the form of text. It is very common on many instant messaging systems such as Skype.

2.1.4 Junk Fax

Junk fax is a means of marketing via unsolicited advertisements that are sent through fax. So the junk faxes are basically the faxed equivalent of a spam mail. It is a medium of telemarketing and ads.

2.1.5 Unsolicited Text Messages Spam or SMS Spam

This type of spam (SMS) is hard to filter. Due to the low cost fo internet and fast progress in trms of technology, it is now very easily possible to send SMS spams at indispensable amounts using the Internet's SMS portals. It is fast becoming a big challenge that needs to be overcome.

2.1.6 Social Networking Spam

Social Networking spam is targeted for the regular users of the social networking websites such as LinkedIn, Facebook, Google+ or MySpace. It often happens that these users of the social networking web services send direct messages or weblinks that contain embedded links or malicious and spam URLs to other locations on the web or to one another. This is how a social spammer plays his role[5].

2.2 Types of Review Spams

Basically three types of review spams exist[6]. These are:

Type 1 (Untruthful Review Spams): Fictitious positive reviews are rewarded to products in order to promote them and also unreasonable negative reviews are given to the competing products to harm their reputations among the consumers. This is how untruthful reviews mislead the consumers into believing their spam reviews.

Type 2 (Reviews with brand mentions): These spams have only brands as their prime focus. They comment about the manufacturer or seller or the brand name alone. These reviews are biased and can easily b figured out as they do not talk about the product and rather only mention the brand names.

Type 3 (Non-reviews): These reviews are either junk, as in, have no relation with the product or are purely used for advertisement purposes. They have these two forms:

i. marketing purposes, and

ii. irrelevant text or reviews having random write-ups.

	Positive spam review	Negative spam review
Good quality product	1	**2**
Bad quality product	**3**	4
Average quality product	**5**	**6**

Figure 2.1: Types of Spams

From Figure 2.1, we can infer that regions 1 and 4 are not very harmful. Regions 2 and 3 are very damaging for th reputation of a product. Regions 5 and 6 are mildly harmful but do bring about significant losses or profits for a brand or a product[7]. In this thesis, we have basically focussed on identifying these regions that are damaging for the product reputation.

2.3 Types of Spammers

While finding spam review we can find two types of spammer Individual Spammer and Group of Spammer[8]. Their traits are as follows:

1. An individual spammer
    - Different user-ids are used to register several times at a website.
    - They build up a reputation.
    - Either only positive reviews are written about a product or only negative reviews about the competitor's products.
    - They give very high ratings for the target products.
2. A group of spammers
    - To control the sales of a product, the spammers write reviews during the launch time of the product.

- Every spam group member write reviews so that the overall product rating deviation lowers down.
- They divide group in sub-groups and then each of these sub divisions work on different web sites.
- They spam at different time intervals to be careful enough to not get detected.

2.4 Related Work

The opinion spam problem was first formulated by in 2008 by Jindal *et al.*[6] in the context of product reviews. By analyzing several million reviews from the popular Amazon.com, they showed how widespread the problem of fake reviews was. The existing detection methods can be split in the context of machine learning into supervised and unsupervised approaches. Second, they can be split into three categories by their features: behavioral, linguistic or those using a combination of these two. They categorized spam reviews into three categories: non-reviews, brand-only reviews and untruthful reviews. The authors ran a logistic regression classifier on a model trained on duplicate or near-duplicate reviews as positive training data, i.e. fake reviews, and the rest of the reviews they used as truthful reviews. They combined reviewer behavioral features with textual features and they aimed to demonstrate that the model could be generalized to detect non-duplicate review spam. This was the first documented research on the problem of opinion spam and thus did not benefit from existing training databases. The authors had to build their own dataset, and the simplest approach was to use near-duplicate reviews as examples of deceptive reviews. Although this initial model showed good results, it is still an early investigation into this problem.

in 2010, Jindal *et al.*[7] did an early work on detecting review spammers which proposed scoring techniques for the spamicity degree of each reviewer. The authors tested their model on Amazon reviews, which were initially taken through several data preprocessing steps. In this stage, they decided to only keep reviews from highly

active users - users that had written at least 3 reviews. The detection methods are based on several predefined abnormalities indicators, such as general rating deviation, early deviation - i.e. how soon after a product appears on the website does a suspicious user post a review about it or very high/low ratings clusters. The features weights were linearly combined towards a spamicity formula and computed empirically in order to maximize the value of the normalized discounted cumulative gain measure. The measure showed how well a particular ranking improves on the overall goal. The training data was constructed as mentioned earlier from Amazon reviews, which were manually labelled by human evaluators. Although an agreement measure is used to compute the inter-evaluator agreement percentage, so that a review is considered fake if all of the human evaluators agree, this method of manually labelling deceptive reviews has been proven to lead to low accuracy when testing on real-life fake review data. First, Ott *et al.* demonstrated that it is impossible for humans to detect fake reviews simply by reading the text. Second, Mukherjee *et al.* proved that not even fake reviews produced through crowdsourcing methods are valid training data because the models do not generalize well on real-life test data.

Wang *et al.*[9] considered the triangular relationship among stores, reviewers and their reviews. This was the first study to capture such relationships between these concepts and study their implications. They introduced 3 measures meant to do this: the stores reliability, the trustworthiness of the reviewers and the honesty of the reviews. Each concept depends on the other two, in a circular way, i.e. a store is more reliable when it contains honest reviews written by trustworthy reviewers and so on for the other two concepts. They proposed a heterogeneous graph based model, called the review graph, with 3 types of nodes, each type of node being characterized by a spamicity score inferred using the other 2 types. In this way, they aimed to capture much more information about stores, reviews and reviewers than just focus on behavioural reviewer centric features. This is also the first study on store reviews, which are different than product reviews. The authors argue that when looking at product reviews, while it may be suspicious to have multiple reviews from the same

person for similar products, it is ok for the same person to buy multiple similar products from the same store and write a review every time about the experience. In almost all fake product reviews, studies which use the cosine similarity as a measure of review content alikeness, a high value is considered as a clear signal of cheating, since the spammers do not spend much time writing new reviews all the time, but reuse the exact same words. However, when considering store reviews, it is possible for the same user to make valid purchases from similar stores, thus reusing the content of his older reviews and not writing completely different reviews all the time. Wang et al. used an iterative algorithm to rank the stores, reviewers and reviews respectively, claiming that top rankers in each of the 3 categories are suspicious. They evaluated their top 10 top and bottom ranked spammer reviewers results using human evaluators and computed the inter-evaluator agreement. The evaluation of the resulted store reliability score, again for the top 10 top and bottom ranked stores was done by comparison with store data from Better Business Bureaus, a corporation that keeps track businesses reliability and possible consumer scams.

Wang *et al.*[9] observed that the vast majority of reviewers (more than 90% in their study or resellerratings.com reviews up to 2010) only wrote one review, so they have focused their research on this type of reviewers. They also claim, similarly to Feng *et al.*,[10], that a flow of fake reviews coming from a hired spammer distorts the usual distribution of ratings for the product, leaving distributional traces behind. Xie *et al.* observed the normal flow of reviews is not correlated with the given ratings over time. Fake reviews come in bursts of either very high ratings, i.e. 5-stars, or very low ratings, i.e. 1-star, so the authors aim to detect time windows in which these abnormally correlated patterns appear. They considered the number of reviews, average ratings and the ratio of singleton reviews which stick out when looking over different time windows. The paper makes important contributions to opinion spam detection by being the first study to date to formulate the singleton spam review problem. Previous works have disregarded this aspect completely by purging singleton reviews from their training datasets and focusing more on tracking

the activity of reviewers as they make multiple reviews. It is of course reasonable to claim that the more information is saved about a user and the more data points about a users activity exist, the easier it is to profile that user and assert with greater accuracy whether he is a spammer or not. Still, it is simply not negligible that a large percentage of users on review platforms write only one review.

Feng *et al.*[10] published the first study to tackle the opinion spam as a distributional anomaly problem, considering crawled data from Amazon and TripAdvisor. They claim product reviews are characterized by natural distributions which are distorted by hired spammers when writing fake reviews. Their contribution consists of first introducing the notion of natural distribution of opinions and second of conducting a range of experiments that finds a connection between distributional anomalies and the time windows when deceptive reviews were written. For the purpose of evaluation they used a gold standard dataset containing 400 known deceptive reviews written by hired people, created by Ott *et al.* Their proposed method achieves a maximum accuracy of only 72.5% on the test dataset and thus is suitable as a technique to pinpoint suspicious activity within a time window and draw attention on suspicious products or brands. This technique does not solely represent however a complete solution where individual reviews can be deemed as fake or truthful, but simply brings to the foreground delimited short time windows where methods from other studies can be applied to detect spammers.

In 2011, Huang *et al.*[11] used supervised learning and manually labelled reviews crawled from Epinions to detect product review spam. They also added to the model the helpfulness scores and comments the users associated with each review. Due to the dataset size of about 60K reviews and the fact that manual labelling was required, an important assumption was made - reviews that receive fewer helpful votes from people are more suspicious. Based on this assumption, they have filtered out review data accordingly, e.g. only considering reviews which have at least 5 helpfulness votes or comments. They achieved a 0.58 F-Score result using their

supervised method model, which outperformed the heuristic methods used at that time to detect review spam. However, this result is very low when compared with that of more recent review spam detection models. The main reason for this has been the training of the model on manually labelled fake reviews data, as well as the initial data pre-processing step where reviews were selected based on their helpfulness votes. In 2013, Mukherjee *et al.*, made the assumption that deceptive reviews get less votes. But their model evaluation later showed that helpfulness votes not only perform poorly but they may also be abused - groups of spammers working together to promote certain products may give many votes to each others reviews. The same conclusion has been also expressed by Jindal *et al.*[7] in 2010.

Ott *et al.*[12] produced the first dataset of gold-standard deceptive opinion spam, employing crowdsourcing through the Amazon Mechanical Turk. They demonstrated that humans cannot distinguish fake reviews by simply reading the text, the results of these experiments showing an at-chance probability. The authors found that although part-of-speech n-gram features give a fairly good prediction on whether an individual review is fake, the classifier actually performed slightly better when psycholinguistic features were added to the model. The expectation was also that truthful reviews resemble more of an informative writing style, while deceptive reviews are more similar in genre to imaginative writing. The authors coupled the part-of-speech tags in the review text which had the highest frequency distribution with the results obtained from a text analysis tool previously used to analyze deception. Testing their classifier against the gold-standard dataset, they revealed clue words deemed as signs of deceptive writing. However, this can be seen as overly simplistic, as some of these words, which according to the results have a higher probability to appear in a fake review, such as vacation or family, may as well appear in truthful reviews. The authors finally concluded that the domain context has an important role in the feature selection process. Simply put, the imagination of spammers is limited - e.g. in the case of hotel reviews, they tend to not be able to give spatial details regarding their stay. While the classifier scored good results on the gold-standard dataset, once

the spammers learn about them, they could simply avoid using the particular clue words, thus lowering the classifier accuracy when applied to real-life data on the long term.

Mukherjee *et al.*[13] were the first to try to solve the problem of opinion spam resulted from a group collaboration between multiple spammers. The method they proposed first extracts candidate groups of users using a frequent itemset mining technique. For each group, several individual and group behavioural indicators are computed, e.g. the time differences between group members when posting, the rating deviation between group members compared with the rest of the product reviewers, the number of products the group members worked together on, or review content similarities. The authors also built a dataset of fake reviews, with the help of human judges which manually labelled a number of reviews. They experimented both with learning to rank methods, i.e. ranking of groups based on their spamicity score and with classification using SVM and logistic regression, using the labelled review data for training. The algorithm, called GSRank considerably outperformed existing methods by achieving an area under the curve result (AUC) of 95%. This score makes it a very strong candidate for production environments where the community of users is very active and each user writes more than one review. However, not many users write a lot of reviews, there exists a relatively small percentage of "elite" contributing users. So this method would best be coupled with a method for detecting singleton reviewers, such as the method from Wang *et al.*

In 2013, Mukherjee *et al.*[14]questioned the validity of previous research results based on supervised learning techniques trained on Amazon Mechanical Turk (AMT) generated fake reviews. They tested the method of Ott et al. on known fake reviews from Yelp. The assumption was that the company had perfected its detection algorithm for the past decade and so its results should be trustworthy. Surprisingly, unlike Ott et al. which reported a 90% accuracy using the fake reviews generated through the AMT tool, Mukherjee's experiments showed only a 68% accuracy when

they tested Otts model on Yelp data. This led the authors to claim that any previous model trained using reviews collected through the AMT tool can only offer near chance accuracy and is useless when applied on real-life data. However, the authors do not rule out the effectiveness of using n-gram features in the model and they proved the largest accuracy obtained on Yelp data was achieved using a combination of behavioural and linguistic features. Their experiments show little improvement over accuracy when adding n-gram features. Probably the most interesting conclusion is that behavioural features considerably outperform n-gram features alone.

Mukherjee *et al.* built an unsupervised model called the Author Spamicity Model that aims to split the users into two clusters - truthful users and spammers. The intuition is that the two types of users are naturally separable due to the behavioural footprints left behind when writing reviews. The authors studied the distributional divergence between the two types and tested their model on real-life Amazon reviews. Most of the behavioural features in the model have been previously used in two previous studies by Mukherjee*et al.* in 2012 and Mukherjee *et al.* in 2013. In these studies though, the model was trained using supervised learning. The novelty about the proposed method in this paper is a posterior density analysis of each of the features used. This analysis is meant to validate the relevance of each model feature and also increase the knowledge on their expected values for truthful and fake reviews respectively.

Fei *et al.*[15] focused on detecting spammers that write reviews in short bursts. They represented the reviewers and the relationships between them in a graph and used a graph propagation method to classify reviewers as spammers. Classification was done using supervised learning, by employing human evaluation of the identified honest/deceptive reviewers. The authors relied on behavioural features to detect periods in time when review bursts per product coincided with reviewer burst, i.e. a reviewer is very prolific just as when a number of reviews which is higher than the usual average of reviews for a particular product is recorded. The authors

discarded singleton reviewers from the initial dataset, since these provide little behaviour information - all the model features used in the burst detection model require extensive reviewing history for each user. By discarding singleton reviewers, this method is similar to the one proposed by Mukherjee *et al.* in 2012. These methods can thus only detect fake reviews written by elite users on a review platform. Exploiting review posting bursts is an intuitive way to obtain smaller time windows where suspicious activity occurs. This can be seen as a way to break the fake review detection method into smaller chunks and employ other methods which have to work with considerably less data points. This would decrease the computational and time complexity of the detection algorithm.

In 2013, Mukherjee *et al.*[14] made an interesting observation in their study: the spammers caught by Yelps filter seem to have overdone faking in their try to sound more genuine. In their deceptive reviews, they tried to use words that appear in genuine reviews almost equally frequently, thus avoiding to reuse the exact same words in their reviews. This is exactly the reason why a cosine similarity measure is not enough to catch subtle spammers in real life scenarios, such as Yelps.

2.5 Spam Detection Methods

1. Supervised Techniques: Supervised spam detection techniques require labelled review spam data set to identify review spam. Its uses several supervised methods, including SVM, logistic regression, Naive Bayes etc. Standard n-gram text classification methodologies can be used to find negative deceptive review spams with an accuracy of roughly 86%.

2. Unsupervised Techniques: Unsupervised methods refers to the problem of finding hidden patterns in data that is unlabelled. Unsupervised methods include k-means clustering, hierarchical clustering, mixture models, etc.

Three different ways of spam detection in the current times are:

1. Review centric spam detection

- Compare content similarity
- Detect rating spikes
- Detect rating and content outliers. (Reviews that have ratings that defer greatly from the average product ratings)
- Compare multiple sites for average ratings

2. Reviewer centric spam detection
    - Watch early reviews
    - Compare the review ratings given by the same reviewer on products from various other stores
    - Compare review times
    - Detect early remedial actions
3. Server centric spam detection
    - We can maintain log of IP address, time of publishing review, site information, etc.

Chapter 3

Supervised Method

3.1 Automated Approaches to Deceptive Review Spam Detection

3.1.1 Linguistic Characteristics as Features

The linguistic and functional properties of text such as its complexity or average number of words per sentence, number of digits, etc.) are an important feature to be incorporated for review spam classification.

Deceptive reviews contain more words, i.e. more **quantity**. The **complexity** in deceptive reviews is found to be greater than truthful reviews. Truthful reviews must essentially have more number of unique words (**diversity**) than deceptive reviews where the spammers have little knowledge about the product. **Brand** names are mentioned more frequently in deceptive reviews than the truthful ones. **Average word length** is more in case of truthful reviews. **No. of digits** mentioned in truthful reviews is more than deceptive as a reviewer writing a truthful review will have more information about the product and hence more digits will be mentioned[3].

3.1.2 Genre Identification: POS Tagging as a Feature

The distribution of parts of speech count (POS Tags) in texts depicts its genre. Strong linguistic differences have been found between imaginative and informative writings, as depicted in the works of Rayson *et al.* in 2001. Informative texts contain more of **nouns, prepositions, adjectives, determiners and coordinating conjunctions**, while the imaginative texts have more of **pronouns, verbs, adverbs**

and pre-determiners. Also number of **Connectors** such as and/or/however are found more in case of imaginative writing such as found in Review Spams. **Immediacy** or number self-referencing words used are also found more in deceptive writing.

3.1.3 Text Categorisation: N-gram as a Feature

N-grams as a feature helps us model the entire content as well as its context using the Text categorisation method. Thus, we consider UNIGRAMS and BIGRAMS in our N-gram feature sets.

Standard techniques for N-gram text categorization have been used to locate Deceptive Review Spams with approximate accuracy of about 86%.

3.1.4 Sentiment as a Feature

The fake negative reviewers are seen to over-produce terms depicting negative emotions (e.g., horrible, disappointed, etc.) as compared to the truthful reviews. Similarly, fictitious positive reviewers over-produced terms depicting emotions of positiveness (e.g., beautiful, elegant, etc.). Therefore, fake hotel reviewers exaggerate the sentiment.

3.2 Classifiers

Features from the four approaches just introduced, linguistic approach, POS tag, polarity and n-gram, are utilized to train classifiers such as Naive Bayes, Decision Tree and Support Vector Machine (SVM).

3.2.1 Naive Bayes

Based on the Bayes theorem, the Naive Bayesian classifier is assumes independence assumptions among different predictors. It is an easy to build model, having no parameter calculation which is complicated enough, and thus can be easily used for huge datasets in particular. Even though this model is highly simplistic, the Naive

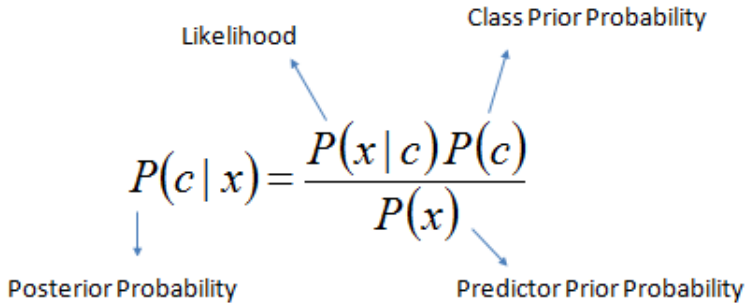
Bayesian classifier performs surprisingly well to be used everywhere and can even outperform the more complicated or sophisticated classification models.

Algorithm:

In Bayes theorem, we ultimately calculate the posterior probability, i.e, P(c |x), from P(c), P(x), and P(x |c).

Here, P(x) is the prior probability, P(x |c) denotes likelihood and P(c) is the class prior probability.

This classifier works on the assumption that value of a feature (x) and its value for a given class will be independent with respect to the values of other feature values. We call this assumption as class conditional independence.



$$P(c \mid \mathbf{X}) = P(x_1 \mid c) \times P(x_2 \mid c) \times \cdots \times P(x_n \mid c) \times P(c)$$

Figure 3.1: Naive Bayes Classifier

where,

P(c |x) : posterior probability of a class given the attributes

P(c) : prior probability of a class

P(x |c) : likelihood, i.e. probability of that feature predictor given a particular class

P(x) : prior probability of feature

Advantages:

1. It works in a single scan, thus it is fast in classification

2. Irrelevant attributes do not affect the classifier performance

3. Examines real data as well as discrete

4. Streaming data is also handled well

Disadvantages:

1. An independence of attributes is assumed

3.2.2 Support Vector Machine

The Support Vector Machine (SVM) classifier is particularly represented by a separating hyperplane. Suppose, we are given labelled training dataset, the algorithm uses supervised learning method thus producing a hyperplane that is the most optimised. This optimised hyperplane then classifies dataset from test set.
Thus, we need to figure out a straight line that separates 2D points in a linear fashion which are distributed among the two classes.

In the process of finding an optimal straight line, if it ends up being close to any

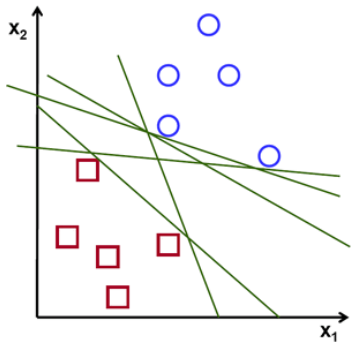


Figure 3.2: Linearly separable set of 2D-points

point, it will be a bad generalisation and might be sensitive to noise and thus incorrect. Thus, our objective will be to be able to get a straight line that is farthest possible from the class points while dividing the class.
The goal of our SVM classifier is to find a hyperplane giving farthest minimum distance between the training class points. We also find something called "margin" in the SVM classifier theory that is twice this separating distance. Finally this hyperplane that

we have found, tends to maximise out training data's "margin".

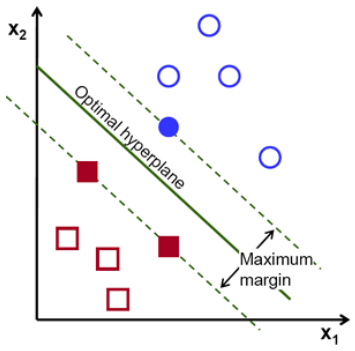


Figure 3.3: Optimal Hyperplane using SVM Classifier

3.2.3 Decision Tree

Decision tree is a classifier that forms a tree structure as a result of classification building and regression models. A decision tree is built in an incremental process by dividing the training data and breaking them into small sets. The classifier finally coms up with the decision tree having decision as well as leaf nodes. Outlook, an example decision node, has branches such as Overcast, Rainy, Sunny, etc. Play, an example leaf node, is a classification point. Root node comes topmost in the decision nodes and decision tree as well. It automatically becomes the best predicator in the classification tree. Categorical as well as numerical data is managed well by decision trees.

Figure 3.4: Decision Tree

Core algorithm for decision tree making, designed by J, R. Quinlan, is called **ID3**. It incorporates a greedy approach and a top down search through the tree's possible branches without backtracking. information Gain algorithm and Entropy methodology are used for making the decision tree using ID3.

Entropy:

Top-down approach is used to build the decision tre starting from root node. Data is partitioned into smaller sets having homogeneous values. ID3 algorithm incorporates Entropy algorithm in order to compute the homogeneity of given data. Entropy becomes zilch is we find that the data is entirely homogeneous. If it is divided in an equal fashion, entropy becomes one.

Information Gain:

A decrease in the entropy value after splitting the dataset on a feature creates information gain. We try to create a decision tree that finds features such that we are able to retrieve the maximum information gain through the most homogeneous branches.

Step 1: The target's entropy value is formulated.

Step 2: We divide the dataset on the basis of our feature attributes while calculating entropy of every branch. Total entropy of the division is obtained by proportional addition. Now the entropy value we have calculated needs to be subtracted from pre-split entropy value. Resulting value obtained is our Information Gain, i.e. the decrease in entropy.

Step 3: We choose the feature that gives us the maximum information gain value and make it our decision node.

Step 4a: If entropy = 0, we term it a leaf node.

Step 4b: If entropy¿0, further splitting needs to be done.

Step 5: We recursively run the ID3 algorithm on decision branches till we classify all th data.

Chapter 4

Proposed Work

4.1 Dataset Collection

4.1.1 Dataset Sources

The area of review spam detection has a very few labelled datasets available. Most of these labellings have either been manually done or is a work of heuristics. We can however, obtain our dataset from various websites mentioned below[12] :

TripAdvisor: TripAdvisor is a website that works particularly in the field of travel and tourism. It contains travel related information and content such as location information and their reviews. There is also a check-in facility where users can mark the places they have just visited. All these facilities are available for no cost. This site has over 60 million members and is one of the largest in the tourism business.
Because of its huge reach and an increased user base, more than 170 million opinions have been collected relating to travel locations, restaurants, motels, hotels, etc., thus making it a very useful resource for our study of review spam. We can scrape data from this website for our data analysis using bots and scripts.

Yelp: Yelp.com is a popular review website which is crowd-sourced and reviews local stores and brands. Here, users can also interact with one another just like in social networking sites. It is more popular in metropolitan areas as a review site. Here, users can rate products or services such as restaurants, mobiles, etc. Star

ratings between 1 to 5 can be given by users. After that they could descriptively write about a product or service. Also, users could check-in, just like Tripadvisor.com, into a restaurant, hotel or a location that they are visiting. Yelp gets about 132 million visitors on a monthly basis and about a total of half a billion reviews.
Although, Yelp does not give away datasets to the public, we can scrape user information and reviews from their website. hough Yelp does not provide its dataset publicly, the reviews and user information can be scraped from the site itself. Bots and scripts can be used to scrape the data as they are allowed with low security so as to get more penetration in search engine results.

Amazon Mechanical Turk: Amazon Mechanical Turk is a service provider that provides on-demand facilities to people. The "requesters", or the general audience, can post assignments in the website which are known as HITS, meaning: Human Intelligence Tasks. "Workers", or the Turkers, as called by the website, need to go through the posted tasks and then complete them in return for a payment.
This website can thus provide us a means to be able to get our spam dataset for research purpose and thus generate fabricated review content as a service task for th Turkers. The task assigned for the workers was to pen down hotel reviews for the mentioned hotels in a manner that they would b accepted and treated as genuine. Thus we get out spam dataset content.

4.1.2 Dataset Description

We compiled a collection of a total of 1600 reviews from the sources mentioned above. These reviews were for 20 Chicago-based hotels. The following are the features of each review:

1. A unique ID for each review for review identification
2. The hotel name about which review has been written
3. The content of the review

4. The polarity of the review, as in whether it portrays positive or a negative sentiment

5. The binary label for depicting whether the review is a spam or not

The data corpus obtained contains the following:

1. 400 truthful reviews (positive). Source: TripAdvisor.com

2. 400 deceptive reviews (positive) Source: Mechanical Turk

3. 400 truthful reviews (negative) Source: Expedia, Yelp, Orbitz, Hotels.com, TripAdvisor and Priceline

4. 400 deceptive reviews (negative) Source: Mechanical Turk

The corpus contains 80 reviews for each of the 20 Chicago-based hotels: Afinia, Amalfi, Allegro, Ambassador, Fairmont, Conrad, Fardrock, Homewood, Hilton, James, Monaco, Hyatt, Intercontinental, Knicker-bocker, Omni, Sharaton, Palmer, Softel, Talbott and Swissotel. These 80 reviews contain 40 spam and 40 non-spam reviews. Each of those 40 reviews have 20 positive and 20 negative reviews [12].

This dataset becomes useful for our research for the following reasons:

1. Our data has reviews in equal numbers for each hotel and thus it is a well-balanced dataset.

2. Class imbalance does not exist as we have spam and non-spam reviews in an equal number with each having negative and positive reviews in a balanced number.

3. While obtaining data from th Amazon Mechanical Turk, we ask the Turkers to review in such a fashion that it seems genuine and can be easily accepted as a good and acceptable review by the website.

4. In the process, the AMT Turkers could also view other reviews already written about the same hotel. Thus a manipulated review, similar to earlier written reviews, made the tasks of the AMT workers much simpler. Thus the knowledge base of the AMT worker also increases to write further genuine-sounding reviews.

5. To ensure ingenuity of the genuine reviews being taken, non-5-star ratings were eliminated.

6. Too short reviews or the ones that were too long were removed.

7. The reviews written first time were eliminated.

8. Reviews written in foreign language were removed such that homogeneity is maintained in the dataset and analysis becomes easier.

4.2 Proposed Work

4.3 Feature Collection

4.3.1 Linguistic Characteristics as Features

Table 4.1: Linguistic Features

Feature Number	Linguistic Feature	Description
F1	Quantity	Total Number of Words
F2	Complexity	Avg number of words per sentence
F3	Diversity	Number of Unique words used
F4	Branding	Frequency of brand names used
F5	Avg Word Length	Ratio of number characters to number of words
F6	Digits	Number of digits used

4.3.2 Genre Identification: POS Tagging as a Feature

In our approach to finding deceptive review spams, we examine the relationship existing between genuine and deceptive reviews. We calculate 9 feature values for each review, based on the POS tags, namely, noun, pronoun, adjective, adverb, verb, determiner, coordinating conjunctions, prepositions and predeterminers.
These POS tag attributes provide a baseline to compare performances of classification models developed and other automated algorithmic processes.

Table 4.2: POS Features

Feature	POS Tag	Description
F7	NN	Number of Nouns
F8	JJ	Number of Adjectives
F9	PRP	Number of Prepositions
F10	DT	Number of Determiners
F11	VB	Number of Verbs
F12	RB	Number of Adverbs
F13	PR	Number of Pronouns
F14	CC	Number of Connector Words
F15	IMM	Number of first person pronouns

4.3.3 Text Categorisation: N-gram as a Feature

Steps:

1. For incorporating N-grams as a feature, we consider unigram and bigram as our feature sets, with the N-grams in lower case as well as unstemmed.

2. We maintain a dictionary for our unigrams and bigram features obtained from the training dataset.

3. Now, from the test site, each review taken is then split into the corresponding N-grams. For each N-gram, its corresponding score is checked.

4. The score is based on either presence in a spam/non-spam set, or its absence, taken in 1s and 0s in the respective cases.

5. Finally, we calculate the total scores to get an idea whether the test review is more similar with spam set or the non-spam set to be able to figure out whether it is genuine or fake.

Now, this score is used to model our classification dataset.

4.3.4 Sentiment as a Feature

Deceptive reviews have been found to contain a greater percentage of words showing positive sentiments than positive genuine reviews. Similarly deceptive negative reviews contain more negative terms than genuine negative reviews[2] [4].

Table 4.3: N-Gram Features

Feature Number	n-gram feature	Description
F16	SpamHitScore	Score indicating how much the words of a review are similar to the spam reviews
F17	NonSpamHitScore	Score indicating how much the words of a review are similar to the spam reviews

$$totss(r) = \frac{\sum(-1)^{c_n}\, o(w_i)}{dist(w_i, f)}$$

Figure 4.1: Sentiment Score calculation

Steps:

1. Extract features/aspect nouns from each sentence in the review.
2. We find the corresponding sentiment words present in the sentence.
3. Strength of the sentiment word on the feature decreases with the distance from the feature word.
4. We calculate the number of negation words to reverse polarity due to negative words present.
5. Finally the aggregation if all feature scores and then its mean gives us the sentiment score in the range [-1,+1].

Here,

r = review

f = aspect/feature in a sentence

o(w_j): sentiment polarity of a word w_j (+1 or -1)

c_n: no. of negation words in one feature, default = 0

dist(w_j,f) = distance between feature f and word w_j.

totss(r) = total sentiment score of a review

Table 4.4: Polarity Features

Feature Number	Feature	Description
F16	Sentiment Score	Range [-1,+1]

Chapter 5

Results

5.1 Linguistic Features Analysis

Table 5.1: Linguistic Features Analysis Result

Approach	Features Considered	Train data size (in %)	Classifier Used	Accuracy (%)
Linguistic Features	Linguistic features vector	70	Naive Bayes	72.04
			SVM	72.1
			Decision Tree	64.60
		80	Naive Bayes	73.25
			SVM	73.25
			Decision Tree	69.00
		90	Naive Bayes	74.02
			SVM	70.89
			Decision Tree	73.2

This Linguistic features analysis works averagely and the results obtained are presented in Table 5.1. Although, we observe that this analysis is comparable to the classification done manually by human annotators in classifying the same dataset. Ott *et al.* discovered that humans have an accuracy level of less than 60% for the same dataset classification task. When multiple groups were asked to classify the dataset, their concurrence of results was pretty low. Thus, our linguistic features model is in tune with the human intuition in deceptive reviews detection.

5.2 POS Features Analysis

POS Features analysis also gives us an average result but its not as good as the results given by the linguistic analysis. Hence in the next sections, we combine the two methods.

Table 5.2: POS Features Analysis Result

Approach	Features Considered	Train data size (in %)	Classifier Used	Accuracy (%)
POS Features	POS Features vector	70	Naive Bayes	68.6
			SVM	63.8
			Decision Tree	66.6
		80	Naive Bayes	67.75
			SVM	62.25
			Decision Tree	71.11
		90	Naive Bayes	72.89
			SVM	66.52
			Decision Tree	68.5

5.3 N-gram Features Analysis

The results obtained from N-gram text classification is shown in Table 5.3. The accuracy levels obtained is fairly better than the ones obtained from linguistic and POS models. Following observations can be made about the same:

Table 5.3: N-gram Features Analysis Result

Approach	Features Considered	Train data size (in %)	Classifier Used	Accuracy (%)
N-gram Features	N-gram Features vector	70	Naive Bayes	73.33
			SVM	73.65
			Decision Tree	72.6
		80	Naive Bayes	72.7
			SVM	76.11
			Decision Tree	73.62
		90	Naive Bayes	96.5
			SVM	88.5
			Decision Tree	96.65

Table 5.4: N-gram Analysis- II

n-gram	Classifier	Accuracy
Bigram	Naive Bayes	73.5
Bigram	SVM	63.75
Bigram	Decision Tree	73.5
Unigram + Bigram	Naive Bayes	71.1
Unigram + Bigram	SVM	60.01
Unigram + Bigram	Decision Tree	71.83

1. We observe that spammers use a set of words frequently in comparison to the genuine review writers. This property is helpful enough for us to classify spam behaviour. Our initial hypothesis is also proved.

2. We find that spammers and non-spammers may have used similar words, but the frequency of its usage from the word-sets makes a huge difference.

3. We can use the N-gram model in general in all types of scenarios, let alone hotel reviews as the basic idea remains same and this method works well on all types of datasets.

5.4 Sentiment Features Analysis

The sentiment scores definitely bring about an increase in the accuracy obtained when combined with the other features.

5.5 Unified Features Model Analysis

The N-gram classification model had an overfitting characteristic for the data-points. The only feature used by this method was the review text that the spammer used. We collaborate the previous three models: Linguistic features, POS Features and the Sentiment score model in order to be able to provide a more realistic model for spam detection and get reasonably good results from the same as can be viewed in Table 5.6. The accuracy levels obtained were fairly more than most of the work done in this area. We obtain about 92.11 % accuracy level obtained by combining the POS, linguistic, sentiment and the unigram feature vectors.

Table 5.5: Sentiment Features Analysis

Features Used	Classifier	Accuracy
Sentiment Score + Linguistic	Naive Bayes	74.5
Sentiment Score + Linguistic	SVM	72.02
Sentiment Score + Linguistic	Decision Tree	75.8
Sentiment Score + POS	Naive Bayes	72.5
Sentiment Score + POS	SVM	70.02
Sentiment Score + POS	Decision Tree	75.7
Sentiment Score + Ling + POS	Naive Bayes	78.9
Sentiment Score + Ling + POS	SVM	74.5
Sentiment Score + Ling + POS	Decision Tree	76.6

Table 5.6: Unified Features Model Analysis

Features Used	Classifier	Accuracy
Sentiment Score + Ling + Unigram Model	Naive Bayes	91.9
Sentiment Score + Ling + Unigram Model	SVM	88.7.1
Sentiment Score + Ling + Unigram Model	Decision Tree	92.11

Chapter 6

Unsupervised Method

6.1 Dataset Collection

We proposed different methods of supervised learning on TripAdvisor dataset.Now we devise another method to test unsupervised learning to detect review spam. Amazon provides its review data in public interest. The data set is available as categorized in various genres of products. For this analysis, a data set for ***Cell Phones and Electronics products*** was used. The data set has ***78,930 reviews*** with each review described as a key-value pair shown below:

Table 6.1: Amazon Cell Phone Reviews Dataset

Description	Size
Cell Phone reviews (78,930 reviews)	20M

Table 6.2: Example Review Data

Tag	Example
product/productId	e.g amazon.com/dp/B00006HAXW
product/price	price of the product
product/title	title of the product
review/userId	id of the user, e.g. A1RSDE90N6RSZF
review/helpfulness	fraction of users who found the review helpful
review/profileName	name of the user
review/score	rating of the product
review/summary	review summary
review/time	time of the review (unix time)
review/text	text of the review

The dataset had many entries that did not have the user ids. Such entries were removed to maintain consistency in the analysis.

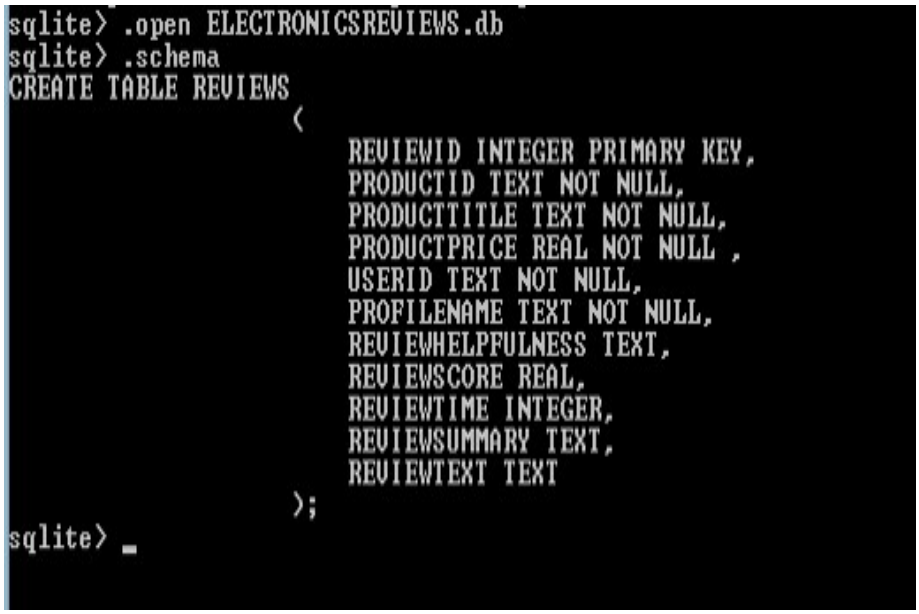


Figure 6.1: Dataset Creation in sqlite

This dataset provides more information than the previous dataset. Apart from review data, it also provides reviewer's as well as the product's information.

6.2 Dataset Analysis

The dataset was analyzed for a number of features described in the next section. However to provide some context to visualize the data, the following charts may be useful.

1. Number of product reviews vs. Number of reviewers

   An interesting observation from the plot is 91% have written 1 review, 99.25% of reviewers have written 3 or less number of reviews. This means the number of people who write a lot of reviews is limited, thus making it easier to red-flag them.

2. Number of product reviews vs. Number of products

   We observe that a large number of products exist that get very few number of reviews and a very small amount of products get high number of reviews.

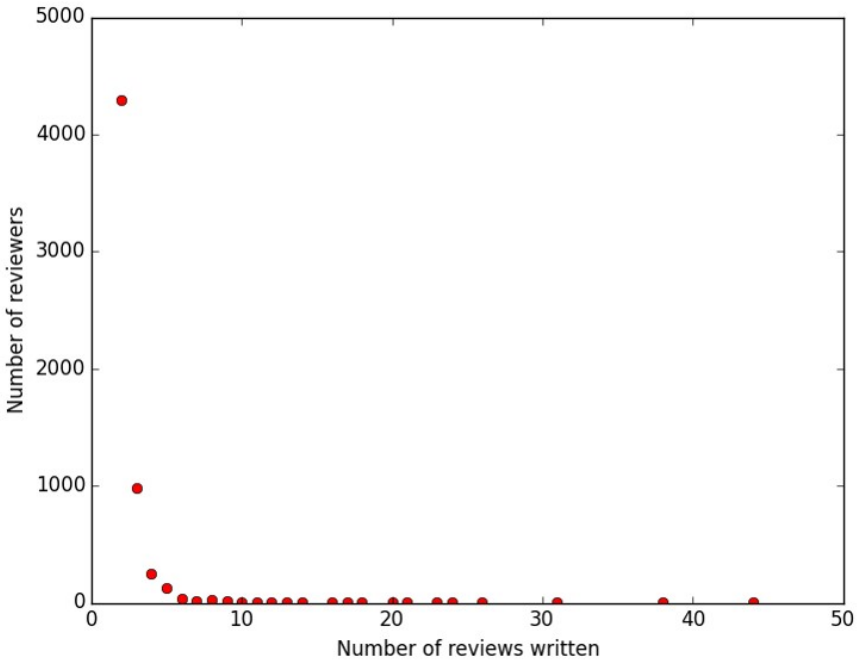


Figure 6.2: Number of product reviews vs. Number of reviewers

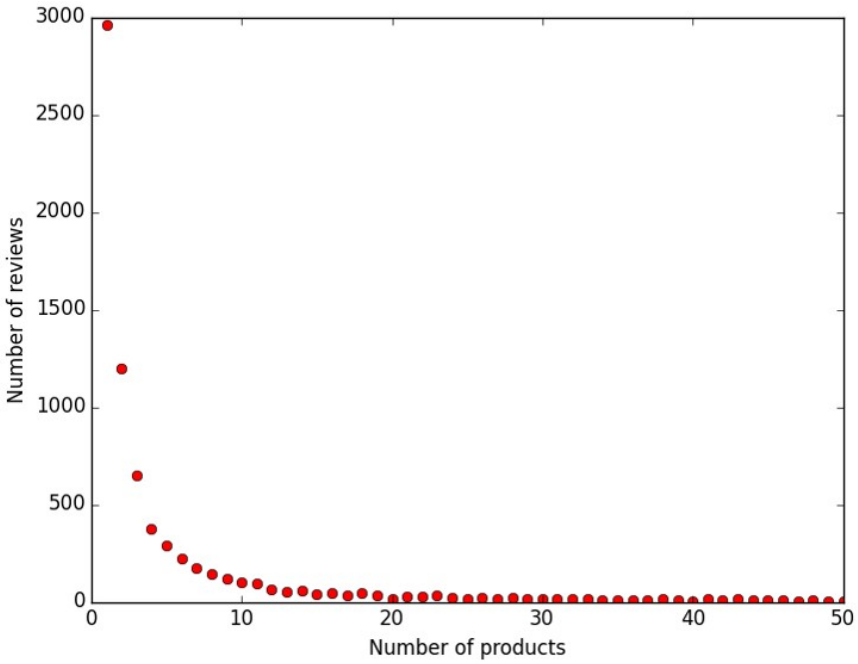


Figure 6.3: Number of product reviews vs. Number of products

3. Product rating vs. Percentage of reviews 60.77% reviews have a rating of 4 and

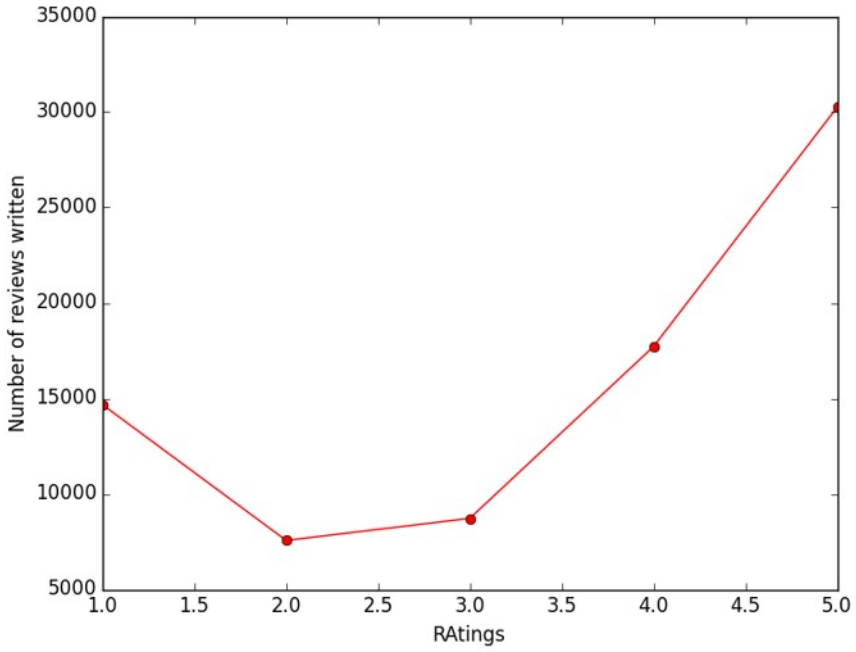


Figure 6.4: Product rating vs. Percentage of reviews

above

4. Number of review pairs vs. Similarity score:

   The Similarity score is the percentage of similar words used. The Similarity score, S is measured out of 100, where a score of hundred means the reviews are identical. The above plot shows the mapping of around 10000 review pairs against their Similarity scores. The plot shows a peak at the middle range values which is to be expected. Beyond the 60 score the plot tapers closely to the x axis. On analysis, around 0.5% of total review pairs have a Similarity score of more than 70. However small this percentage may look, this means around 5500 reviews are near identical copies of previously existing reviews, which in itself is a large number given that the number of products is around 7500.

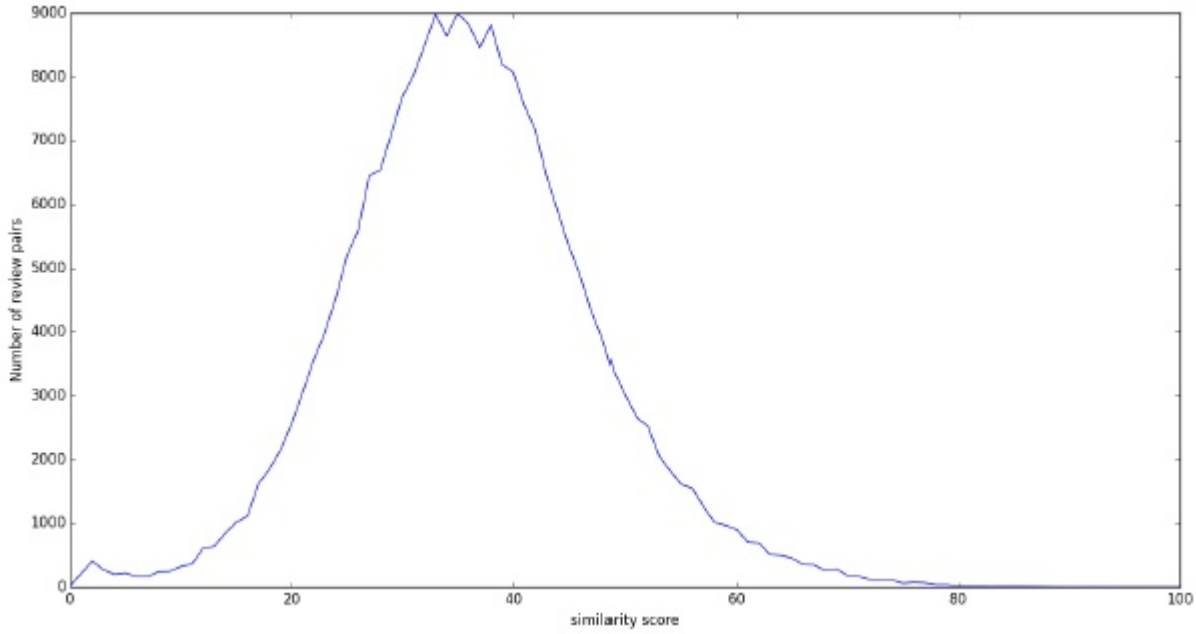


Figure 6.5: Number of review pairs vs. Similarity score

6.3 Feature Vector Generation

The main objective of the unsupervised learning approach is to obtain an effective and suitable feature model for clustering and model building. Information contained in the reviews can be categorised as three main types:

1. Text content of the review
2. Product that is being reviewed
3. Reviewer who writes the review

We thus have three types of features:

1. Review Centric
2. Reviewer Centric
3. Product Centric

As we can infer from the names, review centric features comprise of features related purely to the review text, reviewer centric features contain attributes related to the reviewers and finally, product centric features contain information about the product.

6.3.1 Review Centric Features

F1. Number of review feedbacks

F2. Number of helpful review feedbacks

F3. Percentage of helpful review feedbacks that is received by the review written

F4. Length of the title of the review

F5. Length of the body of the review

(We choose these features since lengthy reviews tend to get higher number of helpful feedbacks and also customers attention. A spammer might use this to their favour.)

F6. Position of the written review among other reviews of that product sorted by date, in ascending

F7. and descending order

We find that reviews written at an early time get more user attention and have a bigger sales impact on th product

F8. Whether a review is first review of that product

F9. Whether a review is that product's only review

Textual features:

F10. Percentage of positive opinion bearing words, e.g.: "beautiful", "great", etc.

F11. Percentage of negative words used in the review, e.g., "bad" and "poor", etc.

F12. Percentage of numerals used,

F13. Number of capitals used

F14. Number of all capitals in the review text

Rating related features:

F15. Rating given for the review

F16. Deviation of this rating from the product rating

F17. Whether a negatively written review was written just after a good review of the given product and

F18. vice versa

6.3.2 Reviewer Centric Features

F19. Ratio of number of reviews written by a reviewer which were first reviews

F20. Ratio of number of times he/she was the only reviewer

F21. Average rating given by a reviewer

F22. Standard deviation in rating given by reviewer

6.3.3 Product Centric Features

F23. Price of a given product

F24. Average rating of a product

F25. Standard deviation in ratings of the reviews on the product

	A	B	C	D	E	F	G	H	I	J	K
1	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11
2	0	0	0	3	142	0	0.018156	0	1	3.333333	2.888889
3	0	0	0	9	31	0	0.02439	0	5	3.333333	2.888889
4	0	0	0	1	27	0	0	0	4	3.333333	2.888889
5	4	5	0.8	3	257	0.012687	0.046269	0.054475	4	3.111111	2.54321
6	0	0	0	4	15	0	0.227273	0.266667	1	3.111111	2.54321
7	0	0	0	3	29	0	0.02963	0.034483	4	3.111111	2.54321
8	0	0	0	2	70	0.007813	0.054688	0.042857	5	3.111111	2.54321
9	0	0	0	8	478	0.007407	0.018107	0.008368	1	3.111111	2.54321
10	1	2	0.5	10	60	0.010169	0.033898	0.016667	3	3.111111	2.54321
11	6	10	0.6	4	35	0	0.016575	0	4	3.111111	2.54321
12	2	4	0.5	10	174	0.003233	0.028017	0.034483	1	3.111111	2.54321
13	1	3	0.333333	7	52	0	0.031873	0.019231	5	3.111111	2.54321
14	2	2	1	7	111	0.015009	0.045028	0.036036	4	3.6	2.24
15	0	0	0	2	152	0.0075	0.05	0.032895	5	3.6	2.24
16	0	0	0	2	25	0	0.04	0	5	3.6	2.24
17	5	8	0.625	3	119	0.008571	0.045714	0.02521	1	3.6	2.24
18	1	2	0.5	9	22	0	0.784314	1	3	3.6	2.24
19	0	0	0	4	36	0	0.031746	0.027778	5	5	0
20	0	2	0	6	17	0	0.018692	0	5	5	0
21	1	1	1	2	29	0.014599	0.029197	0	5	4.5	0.25
22	0	0	0	3	33	0	0.011494	0	4	4.5	0.25
23	1	1	1	3	20	0	0.008621	0	5	5	0

Figure 6.6: A snapshot of the features extracted from the Amazon dataset

6.4 Outlier Spam Detection using k-NN Method

"An **Outlier** is a given observation that deviates from the other observations so much, so as to arouse a suspicion that it was generated by some other mechanism."

Output: List of outliers

Algorithm 1: kNN Algorithm for Outlier Detection

Require: Dataset D, Threshold M, neighbour count K

```
X = getOutlierScores(D,k)
for all p,OutlierScore[p] in X do
   if OutlierScore[p] is greater than or equal to M then
      Add p to L
   end if
end for
getOutlierScores(D,k)
if D != NULL then
   for all p in D do
      S = getKNearestNeighbours(D,p,k)
      for all q in S do
         T = getKNearestNeighbours(D,q,k)
         if p in T then
            Add q to ForwardNNk(p)
            ForwardNNk(p) = ForwardNNk(p) +1
         end if
      end for
   end for
end if
for p in D do
   OutlierScore(p) = 1 - (ForwardNNk(p)/(D-1))
   return [p,OutlierScore(p)]
end for
getKNearestNeighbours(D,p,k)
if D != NULL then
   for all q in D and p !=q do
      Compute dist(p,q)
   end for
end if
sort(dist(p,q))
Add k shortest distant objects from p to NNk(p)
return NNk(p)
```

6.5 Results

Table 6.3: Review Spam Outliers

Total Number of reviews	78930
Number of Spam reviews detected	6064
Percentage of Spam	7.68 %

	A
1	Outliers
2	11
3	19
4	29
5	36
6	39
7	48
8	54
9	57
10	78
11	86
12	95
13	132
14	164
15	172
16	173
17	177
18	184
19	191
20	258
21	326
22	346
23	347

Figure 6.7: Sample of Outliers collected

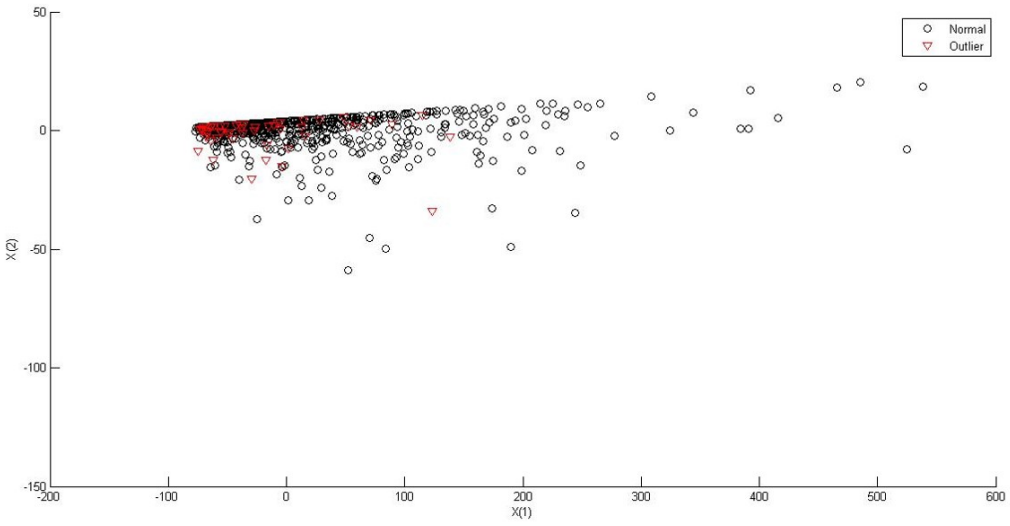


Figure 6.8: Factor Analysis of Outliers Using PCA for d=2

Chapter 7

Conclusion and Future Work

7.1 Supervised Method

The N-gram classification model had an overfitting characteristic for the data-points. The only feature used by this method was the review text that the spammer used. We collaborate the previous three models: Linguistic features, POS Features and the Sentiment score model in order to be able to provide a more realistic model for spam detection and get reasonably good results from the same as can be viewed in Table 5.6. The accuracy levels obtained were fairly more than most of the work done in this area. We obtain about 92.11 % accuracy level obtained by combining the POS, linguistic, sentiment and the unigram feature vectors.

7.2 Unsupervised Method

Based on commonly observed features, it is observed that around 6000 potentially fake entries were obtained. This can be used to build a training once the authenticity of these recognized reviews can denied with a minimum accuracy. When the review was posted also forms crucial part of the analysis. The findings in the bulk analysis can be incorporated into the sequential analysis so that fake reviews can give a red flag as soon as they are submitted. The sentiment of the reviews is also something that can be incorporated in the model. Sites like Amazon, have recently introduced an option that marks verified buyer against the reviews, thus taking a leap in avoiding the impact of opinion spam. The reviews of these verified buyers can be used as a benchmark to demarcate the true reviews from the fake ones.

7.3 Future Work

Just on the basis of the text of the review, the n-gram feature analysis gives a reasonably good result and works pretty effectively in detecting the spam reviews. We observe that the linguistic features as well as the pos model provide a secondary support for our classification model. The combined model gives more reasonable results as it also encompasses the psychological tendency of the spammer. Furthermore, we infer that our spam analysis is incomplete without the reviewer's information. It makes our data much more powerful. Some of the user metadata such as timeframe of writing the reviews, number of written reviews, IP address of the reviewer, age of the reviewer, etc. could be very crucial for our spam analysis and could help in determining fraudulent reviews and spams.

Unfortunately, due to privacy concerns, we do not obtain the user information on the mentioned websites and only those websites can analyse the user data internally. W could also check for the genuine quotient of the text by matching the reviews with information available in the official websites for the given products, such as electronics reviews could be checked against engadget or techcrunch and hotel reviews could be checked against critical reviewers for the same. Nevertheless, we could use this proposed work as a basline for further improvements in this research area.

Bibliography

[1] Kyung-Hyan Yoo and Ulrike Gretzel. Comparison of deceptive and truthful travel reviews. *Information and communication technologies in tourism 2009*, pages 37–47, 2009.

[2] Qingxi Peng and Ming Zhong. Detecting spam review through sentiment analysis. *Journal of Software*, 9(8):2065–2072, 2014.

[3] C Harris. Detecting deceptive opinion spam using human computation. In *Workshops at AAAI on Artificial Intelligence*, 2012.

[4] Myle Ott, Claire Cardie, and Jeffrey T Hancock. Negative deceptive opinion spam. In *HLT-NAACL*, pages 497–501, 2013.

[5] M Daiyan, SK Tiwari, and MA Alam. Mining product reviews for spam detection using supervised.

[6] Nitin Jindal and Bing Liu. Opinion spam and analysis. In *Proceedings of the 2008 International Conference on Web Search and Data Mining*, pages 219–230. ACM, 2008.

[7] Ee-Peng Lim, Viet-An Nguyen, Nitin Jindal, Bing Liu, and Hady Wirawan Lauw. Detecting product review spammers using rating behaviors. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, pages 939–948. ACM, 2010.

[8] Manali S Patil and AM Bagade. Online review spam detection using language model and feature selection. *International Journal of Computer Applications (0975–8887) Volume*, 2012.

[9] Sihong Xie, Guan Wang, Shuyang Lin, and Philip S Yu. Review spam detection via temporal pattern discovery. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 823–831. ACM, 2012.

[10] Song Feng, Longfei Xing, Anupam Gogar, and Yejin Choi. Distributional footprints of deceptive product reviews. In *ICWSM*, 2012.

[11] Fangtao Li, Minlie Huang, Yi Yang, and Xiaoyan Zhu. Learning to identify review spam. In *IJCAI Proceedings-International Joint Conference on Artificial Intelligence*, volume 22, page 2488, 2011.

[12] Myle Ott, Yejin Choi, Claire Cardie, and Jeffrey T Hancock. Finding deceptive opinion spam by any stretch of the imagination. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 309–319. Association for Computational Linguistics, 2011.

[13] Arjun Mukherjee, Bing Liu, and Natalie Glance. Spotting fake reviewer groups in consumer reviews. In *Proceedings of the 21st international conference on World Wide Web*, pages 191–200. ACM, 2012.

[14] Arjun Mukherjee, Abhinav Kumar, Bing Liu, Junhui Wang, Meichun Hsu, Malu Castellanos, and Riddhiman Ghosh. Spotting opinion spammers using behavioral footprints. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 632–640. ACM, 2013.

[15] Geli Fei, Arjun Mukherjee, Bing Liu, Meichun Hsu, Malu Castellanos, and Riddhiman Ghosh. Exploiting burstiness in reviews for review spammer detection. In *ICWSM*. Citeseer, 2013.

[16] Xia Hu, Jiliang Tang, Huiji Gao, and Huan Liu. Social spammer detection with sentiment information.