

OPINION SPAM: FAKE ONLINE REVIEWS DETECTION

**A project report submitted for the partial fulfilment
of the Bachelor of Technology Degree in Computer Science & Engineering
under the Maulana Abul Kalam Azad University of Technology.**

BY

ATHARVA SHRIKANT
(ROLL NO: 10400114042 , REGISTRATION NO: 141040110042)

&

DEBOLEENA CHANDA
(ROLL NO: 10400114057 , REGISTRATION NO: 141040110057)

Under the Guidance of:

Prof. SUKANYA MUKHERJEE
Department of Computer Science and Engineering

For the Academic Year 2014 - 2018.



Institute of Engineering & Management
Y-12, Salt Lake, Sector-V, Kolkata-700091
Affiliated To:



Maulana Abul Kalam Azad University of Technology, West Bengal
formerly known as **West Bengal University of Technology**

BF 142, Sector 1, Salt Lake City, Kolkata-700064

Abstract

The rapid upsurge in the number of e-commerce web sites has made the internet, an extensive source of product reviews. It is now a common practice for e-commerce web sites to enable their customers to write reviews of products that they have purchased. Such reviews provide valuable sources of information on these products. They are used by potential customers to find opinions of existing users before deciding to purchase a product. They are also used by product manufacturers to identify problems of their products and to find competitive intelligence information about their competitors.

Unfortunately, this importance of reviews also gives good incentive for spam, which contains false positive or malicious negative opinions. 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 make an attempt to study review spam and spam detection by building classifier using semi-supervised self-training techniques and compare their accuracy with supervised learning techniques using Decision Trees and Naïve Bayes classifier as base learners. These classifiers will be used on Yelp.com review dataset to classify them as recommended or non-recommended. For the purpose of this project, we would be assuming Yelp classification as pseudo ground truth.

Keywords: Review Spam, Spam Detection, Opinion Spam, Semi-Supervised Learning.

ACKNOWLEDGEMENT

We should like to take this opportunity to extend our gratitude to the following revered persons without whose immense support, completion of this project wouldn't have been possible.

We are sincerely grateful to our advisor and mentor **Prof. Sukanya Mukherjee** of the department of **Computer Science and Engineering**, IEM Kolkata, for her constant support, significant insights and for generating in us a profound interest for this subject that kept us motivated during the entire duration of this project.

We would also like to express our sincere gratitude to **Prof. Dr. Satyajit Chakrabarti** (Director, IEM), **Prof. Dr. Amlan Kusum Nayak** (Principal, IEM) and **Prof. Dr. Debika Bhattacharyya**, HOD of **Computer Science and Engineering** and other faculties of Institute of Engineering & Management, for their assistance and encouragement.

Last but not the least, we would like to extend our warm regards to our families and peers who have kept supporting us and always had faith in our work.

Atharva Shrikant

Reg. No: 141040110042

Dept. of Computer Science & Engineering
Institute of Engineering & Management

Deboleena Chanda

Reg. No: 141040110057

Dept. of Computer Science & Engineering
Institute of Engineering & Management

TABLE OF CONTENTS

TOPIC	PAGE NUMBER
CHAPTER 1. INTRODUCTION	1
1.1 Motivation	2
1.2 Challenges in Review Spam Detection	3
1.3 Objectives	4
1.4 Organisation	4
CHAPTER 2. BACKGROUND	5
2.1 Literature Review	5
2.1.1 Types of Spam	5
2.1.2 Types of Review Spam	6
2.1.3 Types of Spammers	7
2.2 Related Work	7
2.3 Spam Detection Methods	12
CHAPTER 3. PROPOSED METHOD	14
3.1 Collection of Data	14
3.2 Preprocessing of Data	15
3.2.1 Cleaning of Data	15
3.2.2 Preprocessing of text reviews	15
3.2.3 Calculating Behavioral Dimensions	16
3.3 Sampling	17
3.3.1 Simple Random Sampling	18
3.3.2 Stratified Sampling	19

3.4 Machine Learning Techniques	21
3.4.1 Classifier	21
3.4.2 Semi-Supervised Setting	24
3.5 Plan	25
CHAPTER 4. EXPERIMENTAL RESULTS AND ANALYSIS	25
4.1 Result Evaluation	31
4.1.1 Critical Evaluation of Naïve Bayes Experiment	31
4.1.2 Critical Evaluation of Decision Tree Experiment	31
CHAPTER 5. CONCLUSIONS	32
References	32

List of Figures

Figure 1.1 Review Websites	2
Figure 1.2 An example practice of review spam	3
Figure 2.1 Types of Spam	8
Figure 3.1 Collection of Yelp Data	15
Figure 3.2 Preprocessed Reviews	15
Figure 3.3 Word Cloud of the reviews	16
Figure 3.4 A visual representation of sampling process	18
Figure 3.5 A visual representation of selecting a random sample using stratified sampling	20
Figure 3.6 Naïve Bayes Classifier	22
Figure 3.7 Decision Trees	23

Figure 4.1 Semi-supervised vs Supervised using Naïve Bayes (Simple Random Sampling)	29
Figure 4.2 Semi-supervised vs Supervised using Naïve Bayes (Simple Random Sampling)	29
Figure 4.3 Semi-supervised vs Supervised using Decision Trees (Stratified Sampling)	30
Figure 4.4 Semi-supervised vs Supervised using Decision Trees (Stratified Sampling)	30

List of Tables

Table 1.1 Challenges in Review Spam Detection	3
Table 2.1 Types of Spam	6
Table 3.1 List of Notations	16
Table 4.1 Semi-supervised vs Supervised using Naïve Bayes (Simple Random Sampling)	27
Table 4.2 Semi-supervised vs Supervised using Naïve Bayes (Simple Random Sampling)	27
Table 4.3 Semi-supervised vs Supervised using Decision Trees (Stratified Sampling)	28
Table 4.4 Semi-supervised vs Supervised using Decision Trees (Stratified Sampling)	28

CHAPTER – 1

INTRODUCTION

The Web has dramatically changed the way that people express themselves and interact with others. They can now post reviews of products at merchant sites (e.g., amazon.com) and express their views in blogs and forums. It is now well recognized that such user generated contents on the Web provide valuable information that can be exploited for many applications. In this paper, we focus on customer reviews of products, which contain information of consumer opinions on the products, and are useful to both potential customers and product manufacturers.

What is a 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. 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.

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

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 products, are called negative deceptive review spams.

1.1 Motivation

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 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.

The screenshot shows a web interface for a crowdsourcing task. At the top, there is a timer set to 60 minutes and two buttons: 'Accept HIT' and 'Skip HIT'. Below this, a box contains the task details: 'Write Product Reviews 25-50 Words', 'Requester: Mike Bayard', and 'Qualifications Required: HIT approval rate (%) is not less than 95'. The main heading is 'Write a Positive 5/5 Review for Product on Website'. Underneath, it says 'Positive review writing.' followed by a bulleted list of instructions: use best grammar, give a 100% rating, keep entry between 25-50 words, write as if you own the product, tell a story of why you bought it, thank the website, and mark negative reviews as 'not helpful'. Below the list, there is a section titled 'Instructions:' which provides a detailed explanation of the task, including the need to create an account on the product's website and use a specific email address for posting reviews.

Figure 1.2: An example practice of review spam

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 Objectives

The main objective of our project is to build classifiers using Semi-Supervised learning methods. We will then use this classifier to identify “fake” restaurant reviews posted on Yelp. Yelp is a website which publishes crowd-sourced reviews about local businesses including restaurants. Yelp uses its own proprietary algorithm for filtering “fake” reviews. For the purpose of this project, we would be assuming Yelp classification as pseudo ground truth. Semi-supervised learning is a class of supervised learning tasks and techniques that also make use of unlabeled data for training - typically a small amount of with a large amount of unlabeled data. Supervised learning methods are effective when there are sufficient labeled instances to construct classifiers. Labeled instances are often difficult, expensive, or time consuming to obtain, because they require empirical research. When it comes to restaurant reviews, we have a large supply of unlabeled data. Often semi supervised learning achieves a better accuracy than supervised learning which is only trained on the labeled data.

There are various approaches that can be used for semi-supervised learning. These include Expectation Maximization, Graph Based Mixture Models, Self-Training and Co-Training methods. In our project, we will be focusing on applying the Self-Training approach to Yelp’s reviews. In self-training, the learning process employs its own predictions to teach itself. An advantage of self-training is that it can be easily combined with any supervised learning algorithm as base learner.

We will be using three different supervised learning methods - Naïve Bayes, Decision Trees and Logistic Regression as base learners. We would then be comparing the accuracy of each of the semi-supervised learning methods with its respective base learner. The base learners would be using both behavioral and linguistic features.

1.4 Organisation

CHAPTER 1. INTRODUCTION: Gives a brief introduction on what is review spam, what is being done and why this topic has been chosen and challenges to review spam detection.

CHAPTER 2. BACKGROUND: In this section we shall discuss a literature survey of projects done on similar topics and how the authors tried to achieve their objective.

CHAPTER 3. PROPOSED METHOD: In this section, the methods of how to go about the entire technique is described.

CHAPTER 4. EXPERIMENTAL RESULTS AND ANALYSIS: This section produces a report of how our framework is performing.

CHAPTER 5. CONCLUSION: The limitations of our framework and future scope of the work i.e. where and how we can improve this technique to make it more suitable.

CHAPTER – 2

BACKGROUND

2.1 Literature Review

2.1.1 Types of Spam

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.

Comment Spam

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

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.

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.

Unsolicited Text Messages Spam or SMS Spam

This type of spam (SMS) is hard to filter. Due to the low cost for internet and fast progress in terms 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.

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 web links that

contain embedded links or malicious and spam URLs to other locations on the web or to one another. This is how a social spam

spammer plays his role

2.1.2 Types of Review Spam

Basically three types of review spams exist. 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 be 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 the 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. In this thesis, we have basically focussed on identifying these regions that are damaging for the product reputation.

2.1.3 Types of Spammers

While finding spam review we can find two types of spammer Individual Spammer and Group of Spammer. 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.

2.2 Related Work

Extensive studies (Mukherjee et al., 2013; Liu et al., 2013) have been done on determining the effectiveness of existing research methods in detecting real-life fake reviews on a commercial website like Yelp and in trying to emulate Yelp's fake review filtering algorithm.

Apart from this, Liu et al., (2013) proposed a novel model to detect opinion spamming in a Bayesian framework and model the spamicity of reviewers by identifying certain behavioral features. The key motivation is based on the hypothesis that opinion spammers differ from others on behavioral dimensions.

Research has also been done (Jafar et al., 2015) in the application of semi supervised learning to a pool of unlabeled data and augmenting performance of supervised learning algorithm. They have studied the semi-supervised self-training algorithm with decision trees as base learners.

The opinion spam problem was first formulated by in 2008 by Jindal et al. in the context of product reviews. By analysing 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. 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 pre-processing 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. 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 likeness, 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. 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., 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 user's 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. 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. 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 other's reviews. The same conclusion has been also expressed by Jindal et al. 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 analyse 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 item set 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. 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. 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.

2.3 Spam Detection Methods

2.3.1 Supervised Techniques

Supervised spam detection techniques require labelled review spam data set to identify review spam. It 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%.

2.3.2 Unsupervised Techniques

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

2.3.3 Semi-supervised Techniques

Semi-supervised learning is a learning paradigm concerned with the study of how computers and natural systems such as humans learn in the presence of both labeled and unlabeled data. Traditionally, learning has been studied either in the unsupervised paradigm (e.g., clustering, outlier detection) where all the data are unlabeled, or in the supervised paradigm (e.g., classification, regression) where all the data are labeled. The goal of semi-supervised learning is to understand how combining labeled and unlabeled data may change the learning behavior, and design algorithms that take advantage of such a combination. Semi-supervised learning is of great interest in machine learning and data mining because it can use readily available unlabeled data to improve supervised learning tasks when the labeled data are scarce or expensive. Semi-supervised learning also shows potential as a quantitative tool to understand human category learning, where most of the input is self-evidently unlabelled. Some popular semi-supervised learning models, includes self-training, mixture models, co-training and multiview learning, graph-based methods, and semi-supervised support vector machines. For each model, we discuss its basic mathematical formulation. The success of semi-supervised learning depends critically on some underlying assumptions. We emphasize the assumptions made by each model and give counterexamples when appropriate to demonstrate the limitations of the different models.

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 rating.
 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
- 3 We can maintain log of IP address, time of publishing review, site information, etc.

CHAPTER – 3

PROPOSED METHOD

Extensive studies have already been done on detecting spam using supervised learning techniques. Mukherjee et al., (2013) have built upon this by using Yelp’s classification of the reviews as pseudo ground truth. Additionally, Li et al., (2011) have used semi supervised co-training on manually labeled dataset of fake and non-fake reviews. For our project, we will be focusing on applying semi-supervised self-training to Yelp’s reviews by using Yelp’s classification as pseudo ground truth. Our approach is inspired from the above two state of art research on review classification.

We aim to come up with a new solution that will help increase the performance of semi-supervised approach – the idea being that semi-supervised learning methods could improve upon the performance of supervised learning methods in the presence of unlabeled data.

To test this hypothesis, we implemented the self-training algorithm using Naïve Bayes, Decision Trees and Logistic Regression as base learners and compared their performance.

3.1 Collection of Data

We built a Python crawler to collect restaurant reviews from Yelp. Reviews were collected for all restaurants in a particular zip code in New York. We collected both the recommended and non-recommended reviews as classified by Yelp. The dataset consists of approximately 40k unique reviews, 30k users and 140 restaurants. The following attributes were extracted:

1. Restaurant Name
2. Average Rating
3. User Name
4. Review Text
5. Rating
6. Date of Review
7. Classification by Yelp (Recommended / Not Recommended)

1	Restaurant	RestaurantID	AvgRating	AuthorName	NumberOfFriends	NumberOfReview	Rating	Review	Date	Class
2	Eataly	eataly-new-york-8	4	Adam B.	99	271	4	Gazillions of reviews on here, so I	5/5/2015	1
3	The Grey Dog	the-grey-dog-new-york-	4	Everlyn N.	0	14	4	The employees are very nice; the	2/1/2014	1
4	Tonyâ€™s Di Napoli	tonys-di-napoli-new-yor	4	Mariela T.	0	2	5	The best italian food in New York	2/17/2015	1
5	Tonyâ€™s Di Napoli	tonys-di-napoli-new-yor	4	Priscilla P.	107	626	3	Nothing fancy or authentic. Very s	10/25/2014	1
6	The Red Cat	the-red-cat-new-york	4	Fran T.	0	43	4	Four of us had a wonderful dinner	7/29/2012	1
7	Eataly	eataly-new-york-8	4	Derek A.	6	20	5	I am not sure if words can even de	9/27/2015	1
8	Hee Korean BBQ Grill	hee-korean-bbq-grill-ne	4	Eric D.	52	23	4	Quick service for lunch and price i	9/1/2015	1
9	Gramercy Tavern	gramercy-tavern-new-yc	4.5	El T.	1	25	5	Gone on both birthday lunch and c	9/26/2015	1
10	Burger & Lobster	burger-and-lobster-new	4.5	Paul S.	0	2	1	Food is ok but I would never eat th	2/5/2016	0
11	District Tap House	district-tap-house-new-y	4	Phoebe K.	135	39	4	Pretty chill place to watch the gam	3/8/2016	1
12	Calle Dao	calle-dao-new-york	4	Ilyana M.	0	3	4	Was there just this past week for c	8/28/2014	1
13	Burger & Lobster	burger-and-lobster-new	4.5	Tae S.	162	398	3	Hmm. Â The three menu place wa	12/4/2015	1
14	L & W Oyster Co.	l-and-w-oyster-co-new-y	4	Tae S.	162	398	4	Hmmm this place was good but no	3/9/2016	1
15	The Grey Dog	the-grey-dog-new-york-	4	Callan B.	19	72	3	Good but the prices are pretty ridi	1/8/2016	1
16	David Burke Fabrick	david-burke-fabrick-new	4	Juho L.	20	7	1	So, I went to David Burke's M	7/29/2015	1
17	B&B Restaurant Corp	b-and-b-restaurant-corp	4.5	Mike R.	12	3	4	So good and so many choices! And	10/26/2010	1
18	Gyu-Kaku Japanese BBQ	gyu-kaku-japanese-bbq-	4	Jessica B.	1	13	5	This was my first time coming her	12/29/2015	1
19	Taco Bandito	taco-bandito-new-york	3.5	David O.	0	4	1	We picked Taco Bandito because i	11/13/2013	1
20	Her Name Is Han	her-name-is-han-new-y	4	Cody L.	0	23	4	The girlfriend and I had a late dinn	1/10/2016	1

Figure 3.1 Collection of Yelp Data.

3.2 Preprocessing of Data

We carried out the following steps during preprocessing:

3.2.1 Cleaning of Data

The data that we collected had lots of duplicate records and the first step was to remove these. Following this, we modified the date field of all the records to ensure that the formatting was consistent.

3.2.2 Preprocessing of text reviews

The first step here was to remove all the Stop Words. Stop Words are words which do not contain important significance to be used in search queries. These words are filtered out because they return vast amount of unnecessary information [8]. Then we converted the text to lower case and removed punctuations, special characters, white spaces, numbers and common word endings. Finally, we created the Term Document Matrix to find similarity between the text reviews.

Preprocessed Reviews

based first visit impressed â guest lunch went small plates â seemed interesting pasta entreeswe shared burratina fava bean dip fried artichokes grilled sardi
came play msg theater service nice quick efficient baba ganoush app penn burger wife lobster roll delicious filling beer menu excellent good craft beer seas
stumbled staying near bygreat little restaurantgood wine full barthe bone marrow delicious muscles

disappointed decor atmosphere great really big potential food okay top authentic whatsoever waiter particularly condescending assumed didnt know basics
stars loud bartender keeps touching olives â yes bare hands handling money cash register credit cards knows else â wearing gloveslots cute guys though goo
food good might â go back line isnt long went today inside minutes playing awful music n n f f hide kids hide daughters cause offending everyone mentione
passed april ate cookshop along friends impressed food drinks service will server attentive friendly ordered appetizers share always great way get real taste
amazed high rating many reviews pickup mess sauce everywhere one piece completely broken others falling apart taste certainly didnt make itmaki special l
place goto taco place downtown â fancy modern authentic deliciousthe pastor lengua chorizo really good â also specials written board ive tried times â dont
found kobe burger place today really hoping live prime beyond standards alas bei ordered kobe burger advertises waygu sadly burger waygu kobe name bur
great new york brunch good service quality food downsides aircon quick turnarounds place long brunch still really good

situation initially plan celebrate last day chinese new year old coworkers friends shangrila th ave advertised lantern festival celebration chinese new year sp
ever gone place get bite friends get home want write review yelp dont remember name restaurant sums le grainne cafe liked nothing place says cant wait bi

Figure 3.2 Preprocessed Reviews

Following is the word cloud of the text reviews:



Figure 3.3 Word Cloud of the reviews

3.2.3 Calculating Behavioral Dimensions

Variable/Description	Description
$a; A; r; r_a = (a, r)$	Author a ; set of all authors; a review; review by author a
$f_{\max}(a)$	Maximum number of reviews by author a
$\text{MaxRev}(a)$	Maximum number of reviews posted in a day by an author a
f_{rel}	Length of the review
$f_{\text{Dev}}(r_a)$	Reviewer Deviation for a review r by author a
$*(r_a, p(r_a))$	The $*$ rating of r_a on product $p(r_a)$ on the 5 $*$ rating scale
f_{cs}	Maximum content similarity for an author
$\text{cosine}(r_i, r_j)$	Cosine similarity between review i and j

Table 3.1 List of Notations

Using the attributes that we extracted, we identified the following four behavioral features that could be used to build our classifier (The notations are listed above).

- **Maximum Number of Reviews (MNR):** This feature computes the maximum number of reviews in a day for an author and normalizes it by the maximum value for our data.

$$f_{MNR}(a) = \frac{MaxRev(a)}{\max_{a \in A}(MaxRev(a))}$$

- **Review Length:** This feature is basically the number of words in each preprocessed text review.

$$f_{rel} = length(r_i)$$

- **Rating Deviation:** This feature finds the deviation of reviewer's rating for a particular restaurant from the average rating for that restaurant (excluding the reviewer's rating) and normalizing it by the maximum possible deviation, 4 on a 5-star scale.

$$f_{Dev}(r_a) = \frac{|*(r_a, p(r_a)) - E[*(r_{a' \neq a}, p(r_a))]|}{4}$$

- **Maximum Content Similarity (MCS):** For calculating this feature, we first computed the cosine similarities for every possible pair of reviews that are given by a particular reviewer. Then, we choose the maximum of these cosine similarities to represent this feature.

$$fcs(a) = \max_{r_i, r_j \in R_a, i < j} cosine(r_i, r_j)$$

3.3 Sampling

Using random sampling, we split our data set into training and testing sets in the ratio of 70:30 respectively. Then we divided the training set such that approximately 60 % of the records were unlabeled and the remaining were labeled. Following this, we used subsets of increasing sizes from the labeled data to train the base learner (Naïve Bayes). To generate the subsets of labeled data, we used both simple random sampling and stratified sampling approaches. The results of these approaches are discussed in the Experiment and Results' section.

In statistics, quality assurance, and survey methodology, sampling is the selection of a subset (a statistical sample) of individuals from within a statistical population to estimate characteristics of the whole population. Two advantages of sampling are that the cost is lower and data collection is faster than measuring the entire population.

The sampling process comprises several stages:

- Defining the population of concern
- Specifying a **sampling frame**, a **set** of items or events possible to measure
- Specifying a **sampling method** for selecting items or events from the frame
- Determining the sample size
- Implementing the sampling plan
- Sampling and data collecting

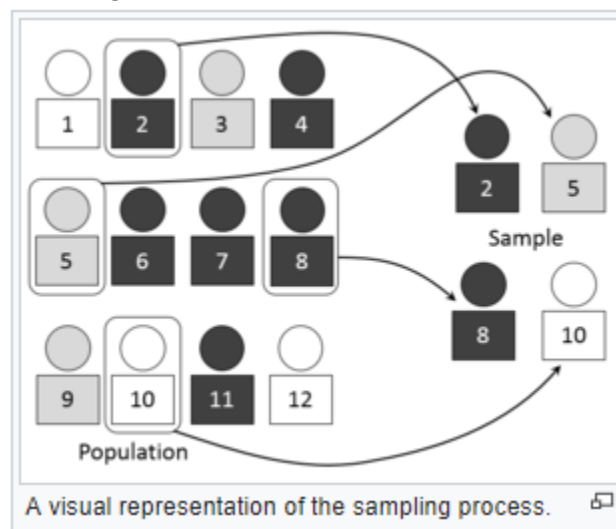


Figure 3.4

3.3.1 Simple Random Sampling

In a simple random sample (SRS) of a given size, all such subsets of the frame are given an equal probability. Each element of the frame thus has an equal probability of selection: the frame is not subdivided partitioned. Furthermore, any given *pair* of elements has the same chance of selection as any other such pair (and similarly for triples, and so on). This minimizes bias and simplifies analysis of results. In particular, the variance between individual results within the sample is a good indicator of variance in the overall population, which makes it relatively easy to estimate the accuracy of results.

SRS can be vulnerable to sampling error because the randomness of the selection may result in a sample that doesn't reflect the makeup of the population. For instance, a simple random sample of ten people from a given country will *on average* produce five men and five women, but any given trial is likely to overrepresent one sex and underrepresent the

other. Systematic and stratified techniques attempt to overcome this problem by "using information about the population" to choose a more "representative" sample.

SRS may also be cumbersome and tedious when sampling from an unusually large target population. In some cases, investigators are interested in "research questions specific" to subgroups of the population. For example, researchers might be interested in examining whether cognitive ability as a predictor of job performance is equally applicable across racial groups. SRS cannot accommodate the needs of researchers in this situation because it does not provide subsamples of the population. "Stratified sampling" addresses this weakness of SRS.

3.3.2 Stratified Sampling

When the population embraces a number of distinct categories, the frame can be organized by these categories into separate "strata." Each stratum is then sampled as an independent sub-population, out of which individual elements can be randomly selected.^[3] The ratio of the size of this random selection (or sample) to the size of the population is called a **sampling fraction**. There are several potential benefits to stratified sampling.

First, dividing the population into distinct, independent strata can enable researchers to draw inferences about specific subgroups that may be lost in a more generalized random sample.

Second, utilizing a stratified sampling method can lead to more efficient statistical estimates (provided that strata are selected based upon relevance to the criterion in question, instead of availability of the samples). Even if a stratified sampling approach does not lead to increased statistical efficiency, such a tactic will not result in less efficiency than would simple random sampling, provided that each stratum is proportional to the group's size in the population.

Third, it is sometimes the case that data are more readily available for individual, pre-existing strata within a population than for the overall population; in such cases, using a stratified sampling approach may be more convenient than aggregating data across groups (though this may potentially be at odds with the previously noted importance of utilizing criterion-relevant strata).

Finally, since each stratum is treated as an independent population, different sampling approaches can be applied to different strata, potentially enabling researchers to use the approach best suited (or most cost-effective) for each identified subgroup within the population.

There are, however, some potential drawbacks to using stratified sampling. First, identifying strata and implementing such an approach can increase the cost and complexity of sample selection, as well as leading to increased complexity of population estimates. Second, when examining multiple criteria, stratifying variables may be related to some, but not to others, further complicating the design, and potentially reducing the

utility of the strata. Finally, in some cases (such as designs with a large number of strata, or those with a specified minimum sample size per group), stratified sampling can potentially require a larger sample than would other methods (although in most cases, the required sample size would be no larger than would be required for simple random sampling).

A stratified sampling approach is most effective when three conditions are met

1. Variability within strata are minimized
2. Variability between strata are maximized
3. The variables upon which the population is stratified are strongly correlated with the desired dependent variable.

Advantages over other sampling methods

1. Focuses on important subpopulations and ignores irrelevant ones.
2. Allows use of different sampling techniques for different subpopulations.
3. Improves the accuracy/efficiency of estimation.
4. Permits greater balancing of statistical power of tests of differences between strata by sampling equal numbers from strata varying widely in size.

Disadvantages

1. Requires selection of relevant stratification variables which can be difficult.
2. Is not useful when there are no homogeneous subgroups.
3. Can be expensive to implement.

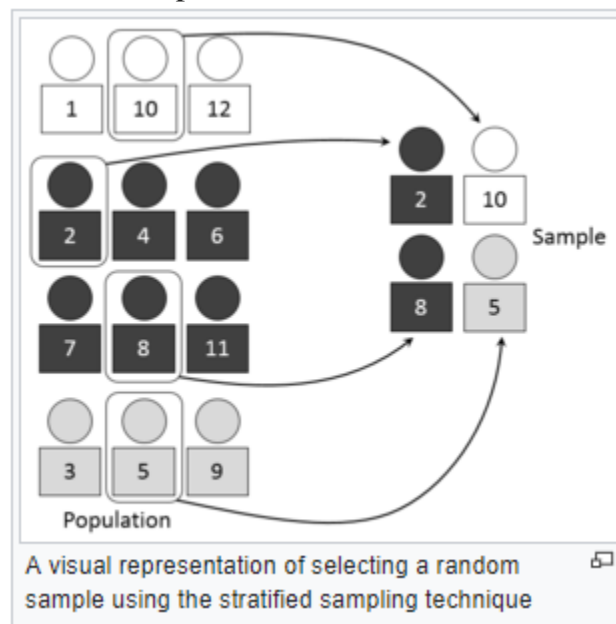


Figure 3.5

3.4 Machine Learning Techniques

In our project we focus on using semi-supervised learning with self-training – a widely used method in many domains and perhaps the oldest approach to semi-supervised learning. We chose to evaluate our classifiers using self-training because it follows an intuitive and heuristic approach. Additionally, the usage of Self-Training allowed us to implement multiple classifiers as base learners (for e.g. Naïve Bayes, Decision Trees, Logistic Regression etc.) and compare their performance.

For the choice of base learners, we had various options. We chose Naïve Bayes, Decision Trees and Logistic regression as our three base learners for the Self-Training algorithm. We chose these options because of the fact that Self-Training requires a probabilistic classifier as input to it. We didn't use non-probabilistic classifiers like Support Vector Machines (SVM) and K-nearest neighbor (k-NN) because of this reason.

We were also considering using co-training as one of our semi-supervised learning approaches. However, Co-Training requires the presence of redundant features so that we can train two classifiers using different features before we finally ensure that these two classifiers agree on the classification for each unlabeled example. For the data-set that we were using, we didn't have redundant features and hence we decided against using Co-Training.

3.4.1 Classifier

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, etc.

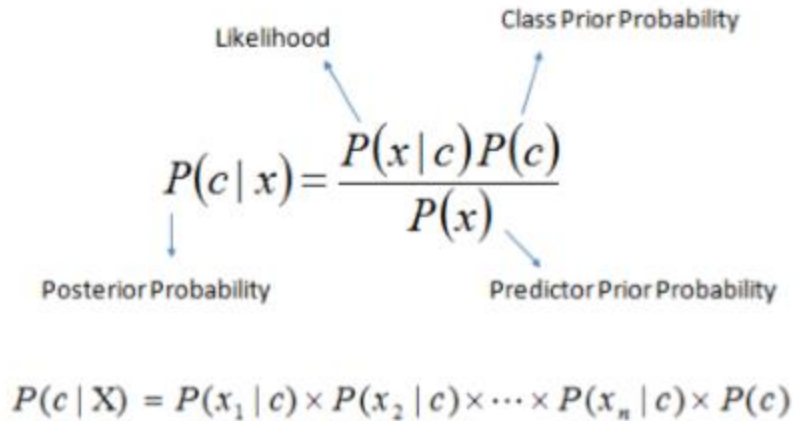
Naive Bayes

Based on the Bayes theorem, the Naive Bayesian classifier 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 Naïve 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



$$P(c | x) = \frac{P(x | c)P(c)}{P(x)}$$

$P(c | X) = P(x_1 | c) \times P(x_2 | c) \times \dots \times P(x_n | c) \times P(c)$

Figure 3.6 Naïve 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 assume

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 comes 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.

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.



Figure 3.7 Decision Trees

Entropy:

Top-down approach is used to build the decision tree 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 zero if 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 the data.

3.4.2 Semi – Supervised setting

In semi-supervised learning there is a small set of labeled data and a large pool of unlabeled data. We assume that labeled and unlabeled data are drawn independently from the same data distribution. In our project, we consider datasets for which $n_l \ll n_u$ where n_l and n_u are the number of labeled and unlabeled data respectively.

First, we use Naïve Bayes as a base learner to train a small number of labeled data. The classifier is then used to predict labels for unlabeled data based on the classification confidence. Then, we take a subset of the unlabeled data, together with their prediction labels and train a new classifier. The subset usually consists of unlabeled examples with high-confidence predictions above a specific threshold value .

In addition to using Naïve Bayes, we are also planning to use Decision Trees and Logistic Regression as base learners. The performance of each of the semi-supervised learning models would then be compared with its respective base learner.

Self – Training

The self-training algorithm wraps around a base classifier and uses its own predictions through the training process. A base learner is first trained on a small number of labeled examples, the initial training set. The classifier is then used to predict labels for unlabeled examples (prediction step) based on the classification confidence. Next, a subset SS of the unlabeled examples, together with their predicted labels, is selected to train a new classifier (selection step). Typically, SS consists of a few unlabeled examples with high-confidence predictions. The classifier is then re-trained on the new set of labeled examples, and the procedure is repeated (re-training step) until it reaches a stopping condition. As a base learner, we employ the decision tree classifier in self-training. The most well-known algorithm for building decision trees is the C4.5 algorithm, an extension of Quinlan's earlier ID3 algorithm. Decision trees are one of the most widely used classification methods. They are fast and effective in many domains. They work well with little or no tweaking of parameters which has made them a popular tool for many domains. This has motivated us to find a semi-supervised method for learning decision trees. Algorithm 1 presents the main structure of the self-training algorithm.

The goal of the selection step in Algorithm 1 is to find a set unlabeled examples with high-confidence predictions, above a threshold TT . This is important, because selection of incorrect predictions will propagate to produce further classification errors. At each iteration the newly-labeled instances are added to the original labeled data for constructing a new classification model. The number of iterations in Algorithm 1 depends on the threshold TT and also on the pre-defined maximal number of iterations,

The outline of the self-training algorithm is given below :

Algorithm 1 : Outline of the Self – Training algorithm

*Initialize: L, U, F, T ; L : Labeled data; U : Unlabeled data;
 F : Underlying classifier; T : Threshold for selection;
 $Iter_{max}$: Number of iterations; $\{P_i\}_{i=1}^M$: Prior probability;
 $t \leftarrow 1$;
while ($U \neq \text{empty}$) and ($t < Iter_{max}$) do
 – $H^{t-1} \leftarrow \text{BaseClassifier}(L, F)$;
 for each $x_i \in U$ do
 – Assign pseudo – label to x_i based on classification confidence
 – Sort Newly – Labeled examples based on the confidence
 – Select a set S of the high – confidence predictions according to $n_i \propto P_i$
 and threshold T // Selection Step
 – Update $U = U - S$; $L = L \cup S$;
 – $t \leftarrow t + 1$
 – Re – Train H^{t-1} by the new training set L
end while
Output: Generate final hypothesis based on the new training set*

3.5 Plan

The main goal of our project was to test the hypothesis that when the number of labeled data is less, semi-supervised learning methods could improve upon the performance of supervised learning methods in the presence of unlabeled data.

To verify this hypothesis, we compared the performance of semi-supervised self-training against its respective base learners. To do this, we performed the following steps:

- We split the available data set into training and testing sets in the ratio of 70:30.
- On the training set, we created labeled data of varying sizes (from 50 to 2000). For the remaining data, we removed the labels and considered it to be the unlabeled data set.
- We then trained the base learners individually on these sets of labeled data and tested it on the test set noting the accuracy.
- Using these base learners, we built the semi-supervised self-training model individually on the sets of labeled data and again tested it on the test set noting the accuracy.
- Finally, we compared the accuracy for the base learners alone and its corresponding semi supervised self-training model and plotted graphs.

One difficulty that we faced while we designed the experiment was that in our dataset, as per Yelp's classification, we had only 11% of data that was classified as spam by Yelp. To ensure that we preserve this ratio between spam vs non spam data while sampling, we decided to use stratified sampling along with simple random sampling. This was done to check if stratified sampling produced any performance improvements.

The following comparisons were made:

- **Semi-Supervised Vs Supervised using Naïve Bayes** – We aim to implement the base learner as Naïve Bayes classifier and use it in the self-training algorithm.
- **Semi-Supervised Vs Supervised using Decision Trees** – We aim to implement the base learner as Decision Tree classifier and use it in the self-training algorithm.

CHAPTER – 4

EXPERIMENTAL RESULTS AND ANALYSIS

Stratified Sampling and Simple Random Sampling

While performing Stratified sampling, we have maintained the same ratio of class labels (recommended vs not recommended) in the labeled dataset as the original dataset

Below is the table of accuracy of supervised and unsupervised techniques:

LabeledData	SupervisedAcc	SemiSupervisedAcc
50	0.4350307	0.7094687
100	0.4734332	0.6080552
150	0.4874830	0.6072888
200	0.8199080	0.6043937
300	0.8314884	0.8171832
500	0.8648672	0.8755960
700	0.8806199	0.8743188
900	0.8722752	0.8632493
1200	0.8761069	0.8744891
1500	0.8765327	0.8755960
2000	0.8717643	0.8726158

Table 4.1 Semi-supervised vs Supervised using Naïve Bayes (Simple Random Sampling)

LabeledData	SupervisedAcc	SemiSupervisedAcc
50	0.5197037	0.5453951
100	0.4367337	0.5084901
150	0.6628065	0.6533549
200	0.8471560	0.8939884
300	0.8617166	0.8939884
500	0.8718495	0.8779802
700	0.8755960	0.8933072
900	0.8801090	0.8801090
1200	0.8541383	0.8660593
1500	0.8853031	0.8853031
2000	0.8768733	0.8807902

Table 4.2 Semi-supervised vs Supervised using Naïve Bayes (Stratified Sampling)

LabeledData	SupervisedAcc	SemiSupervisedAcc
50	0.7315225	0.8915191
100	0.7558753	0.8415191
150	0.6543767	0.8115191
200	0.7998978	0.8293597
300	0.8528610	0.8880279
500	0.8749149	0.8827486
700	0.8701465	0.8884537
900	0.8508174	0.8926260
1200	0.8618869	0.8928815
1500	0.8915191	0.8906676
2000	0.8680177	0.8911785
2500	0.8897309	0.8934775

Table 4.3. Semi-supervised vs Supervised using Decision Tree (Simple Random Sampling)

LabeledData	SupervisedAcc	SemiSupervisedAcc
50	0.6811138	0.8910082
100	0.7875511	0.8618869
150	0.8112228	0.8710659
200	0.8297003	0.8910082
300	0.8251022	0.8910082
500	0.8749149	0.8868358
700	0.8757663	0.8834298
900	0.8855586	0.8899012
1200	0.8579700	0.8905824
1500	0.8473263	0.8908379
2000	0.8719346	0.8910082

Table 4.4. Semi-supervised vs Supervised using Decision Tree (Stratified Sampling)

The following graphs show the results of individual base learners vs. the semi-supervised self-training method for varying labeled data sets:

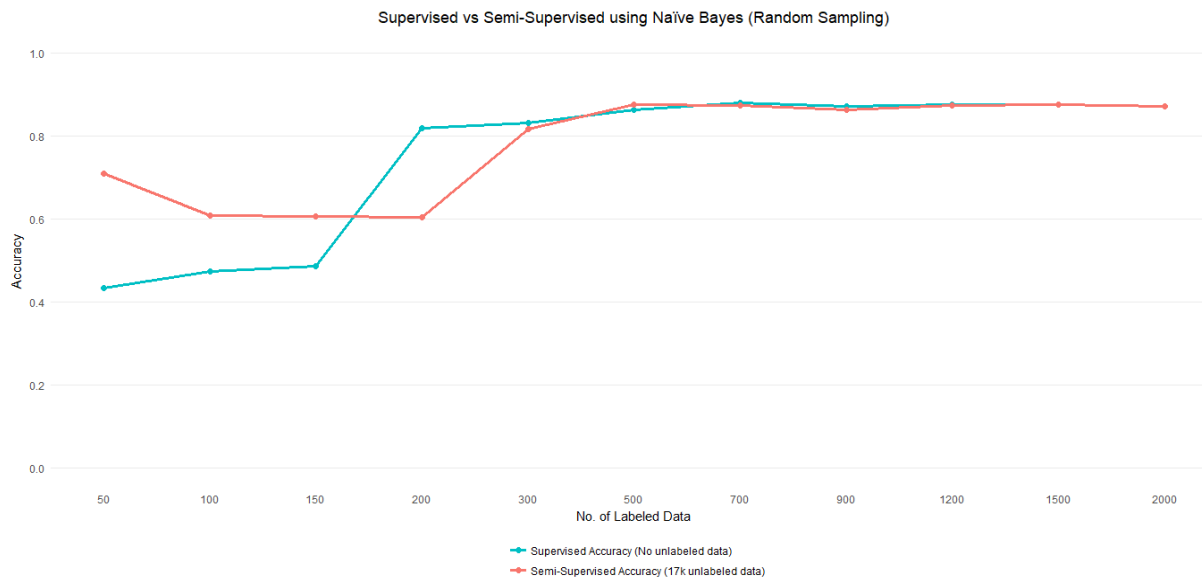


Figure 4.1 Semi-supervised vs Supervised using Naïve Bayes (Simple Random Sampling)

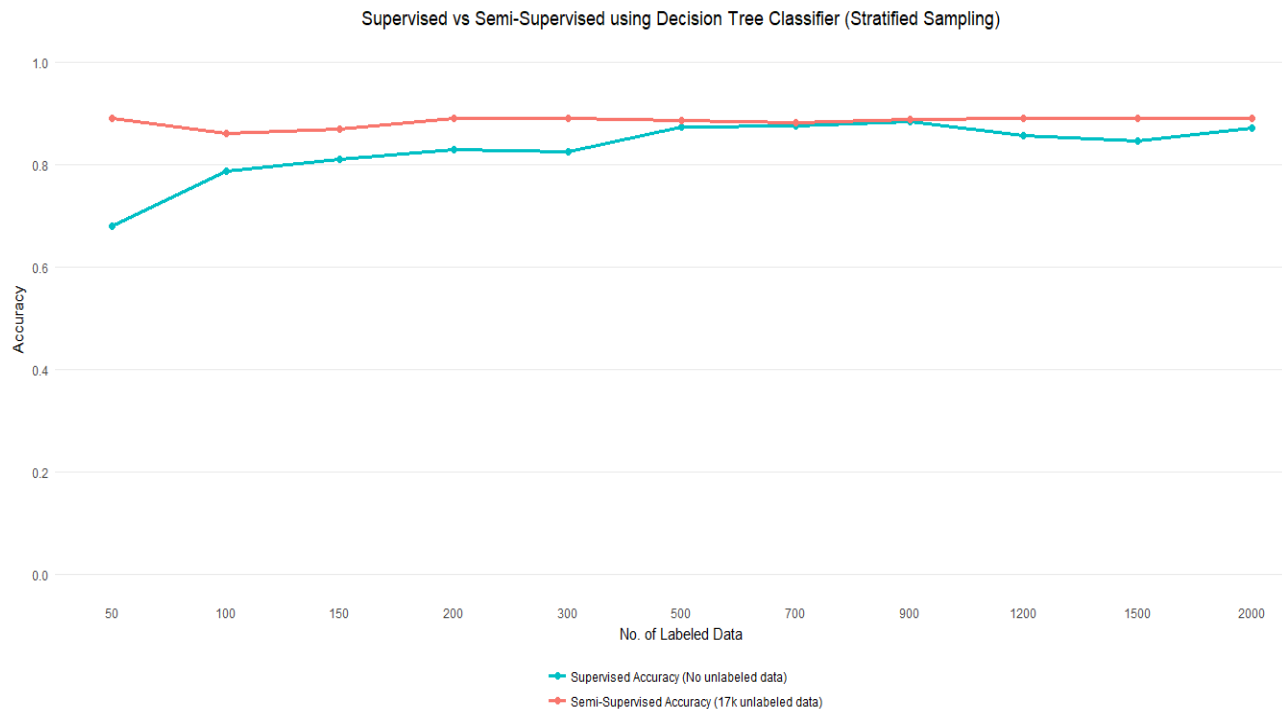


Figure 4.2 Semi-supervised vs Supervised using Naïve Bayes (Stratified Sampling)

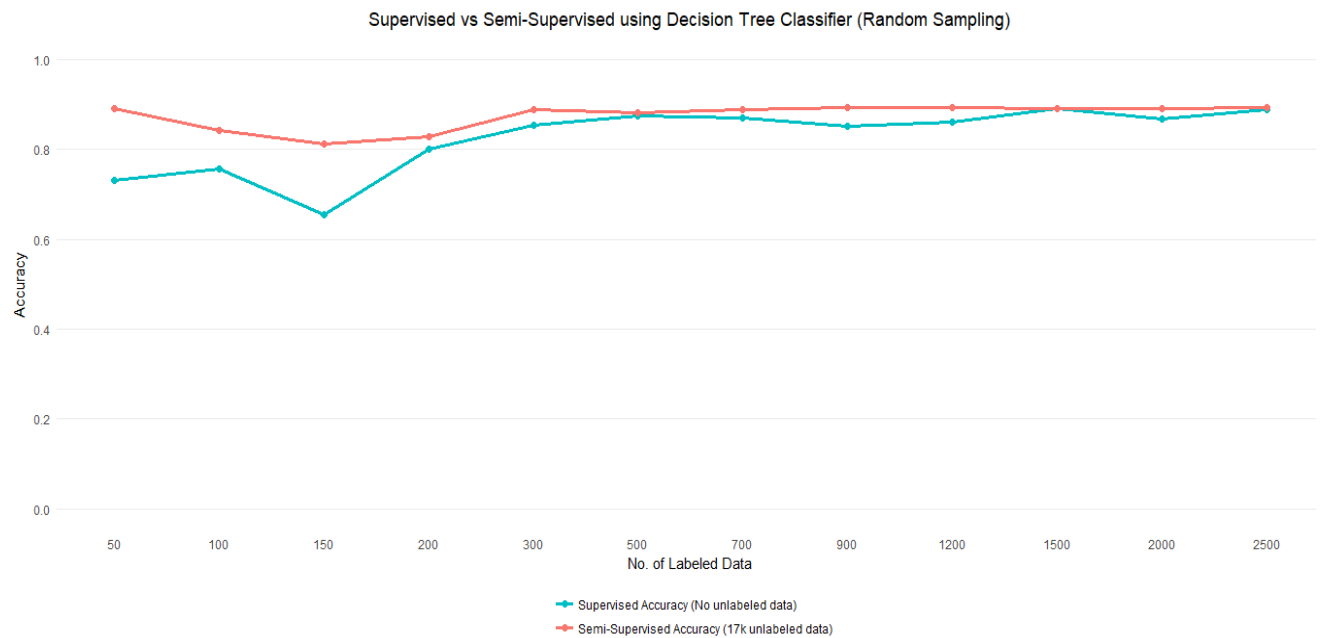


Figure 4.3. Semi-supervised vs Supervised using Decision Tree (Simple Random Sampling)

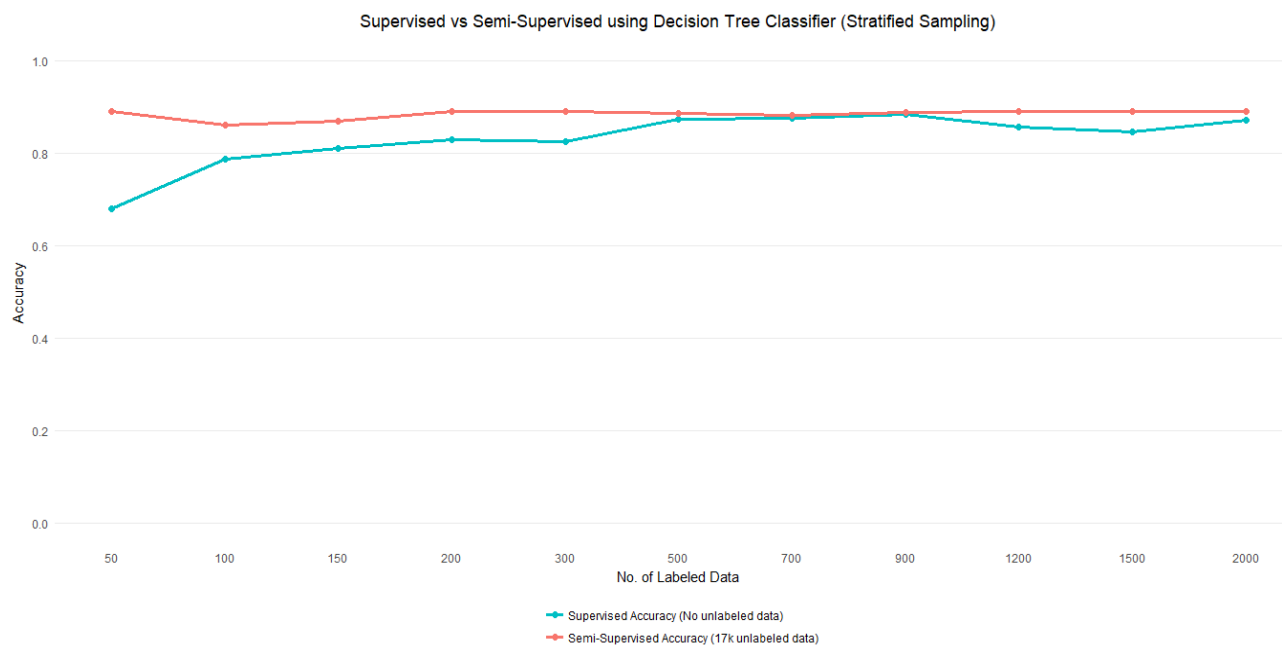


Figure 4.4. Semi-supervised vs Supervised using Decision Tree (Stratified Sampling)

4.1 Result Evaluation

4.1.1 Critical evaluation of the Naïve Bayes experiment

- As the size of the labeled data set increases, accuracy of both the models converged to a stable value (Approximately 86%). Thus, Naïve Bayes performed well for both the supervised and semi-supervised training model.
- When number of labeled data was low, Naïve Bayes with simple random sampling performed better with the semi-supervised model than the supervised approach. For stratified sampling, both the models gave similar accuracy. This is in agreement to our initial hypothesis.
- As we increased the number of labeled data, accuracy for the semi-supervised approach was not always better than the supervised approach. This is a deviation from our initial hypothesis. This might be because Naïve Bayes has the strong assumption that the features are conditionally independent. For our project, it is difficult to interpret the interdependencies between behavioral footprints of the reviewers.

4.1.2 Critical evaluation of the Naïve Bayes experiment

- As the size of the labeled data set increases, accuracy of both the models converged to a stable value (Approximately 89%). Thus, Decision Tree performed well for both the supervised and semi-supervised training model.
- For both simple random and stratified sampling, Decision Tree performed better with the semi-supervised model than the supervised approach. This is in agreement to our initial hypothesis.

CHAPTER – 5

CONCLUSION

Through this project, we learnt that self-training works well when the base learner is able to predict the class probabilities of unlabeled data with high confidence.

Based on the experiments that we performed, we found that in general semi-supervised learning using self-training does improve the performance of supervised learning methods in the presence of unlabeled data.

From the approaches that we tried, we found that semi-supervised self-training using Decision Tree as classifier leads to better selection metric for the self-training algorithm than the Naïve Bayes and Logistic Regression base learners. Thus, Decision tree works as a better classification model for our project.

Since the Decision Tree worked well, we had the idea of implementing Naïve Bayes Tree which is a hybrid of Decision Tree and Naïve Bayes on our data set. Tanha et al., (2015) have conducted a series of experiments which show that Naïve Bayes trees produce better probability estimation in tree classifiers and hence would work well with the self-training algorithm.

References

- [1] Mukherjee, A., Venkataraman, V., Liu, B. and Glance, N. 2013. What Yelp Fake Review Filter might be Doing? ICWSM. (2013).
- [2] Mukherjee, A., Liu, B. and Glance, N. 2012. Spotting fake reviewer groups in consumer reviews. WWW.
- [3] Mukherjee, A., Kumar, A., Liu, B., Wang, J., Hsu, M., Castellanos, M., and Ghosh, R. 2013a. Spotting opinion spammers using behavioral footprints. In Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM.
- [4] Nigam, K., McCallum, A., Thrun, S., and Mitchell, T. Text classification from labeled and unlabeled documents using EM., 2000.

- [5] Tanha, J., Someren, M., and Afsarmanesh, H. "Semi-Supervised Self-Training for Decision Tree Classifiers". *Int. J. Mach. Learn. & Cyber.* (2015): n. pag. Web.
- [6] Tanha, J., Someren, M., and Afsarmanesh, H. "Semi-Supervised Self-Training with Decision Trees: An Empirical Study". In *proceeding of: 3rd IEEE International Conference on Intelligent Computing and Intelligent System*, (2011)
- [7] "Yelp". Wikipedia. N.p., 2016. Web. 10 Apr. 2016.
- [8] "List of English Stop Words - XPO6". XPO6. N.p., 2009. Web. 10 Apr. 2016.
- [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.