Improving Performance of Fake Reviews Detection in Online Review's using Semi-Supervised Learning

Amitkumar B. Jadhav[1], Vijay U. Rathod [2], Dr. Hemantkumar B. Jadhav [3]

[1]ME, Computer Dept., Vishwabharti Academy's College of Engg., Ahmednagar, India
[2]Asst. Prof., Computer Dept., Vishwabharti Academy's College of Engg., Ahmednagar, India
[3]Professor, Computer Dept., Shri Chhatrapati Shivaji Maharaj College of Engg, Ahmednagar, India

---

Abstract - *Ever Since its birth in the late 60's, Internet has been widely and mainly used for interaction purpose only; but over the period of time, its application has changed significantly. Now a days' Internet is no longer use only for communication purpose. Its use is spread over wide variety of applications' and E-Commerce is one of them. The most important part in e-commerce, from consumer perspective is, the reviews associated with products. Most of the people do their decision making, based on these online reviews about products or services. These reviews not only help user to know the product or service thoroughly but also affect user's decision making ability to a great extent and also divert the sentiments about the product positively or negatively. As a result, there have been attempts made, to change the product sentiments positively or negatively by manipulating the online reviews artificially to gain the business benefits. Ultimately, affect the genuine business experience of the user. Therefore in this paper, we have dealt with this particular problem of e-commerce field, specifically online reviews' in particular and sentiment analysis domain as a whole, in general. A ton of work has been already done in this domain since last decade. In this paper, we will see cumulative study of this work and will also see how addition of unlabeled data improve the accuracy in Identifying fake reviews using three different base learner algorithms viz. Naïve Bayes, Decision Tree and Logistic Regression.*

Key Words: **Opinion Mining, Data Analysis, Sentiment Analysis, Opinion Spamming, Fake Review Detection, Semi Supervise learning**

1. INTRODUCTION

Sentiment analysis is the process of extraction of knowledge from the opinions of others. The term is also called as "Opinion Mining". It is area of research that deals with information retrieval and knowledge discovery from text using data mining, natural language processing and machine learning techniques. The knowledge of this analysis can be used for recommendation systems, government intelligence, citation analysis, human-computer interaction and its computer assisted interactivity. The domain of sentiment analysis is vast domain and we have restricted ourselves to the field of online reviews and analysis of the same.

To consult a review and come to a decision, based on sentiment or content of review is very common thing. Before actually buying the particular product or service, people like to read comments, reviews and ratings about that product or service, to get the clear idea about it. Therefor company which sells the product or services exploit this feature to sell more their products by wrongly influencing the potential buyers. In order to do this company hires people who write fake reviews about product for them, also called as spammers and thus process called opinion spamming. This process of opinion spamming can be done in both ways, either to promote your own product by changing sentiment of product positively or to demote competitors' product by changing sentiment of it negatively. According to Mr. Bing Liu, an expert in opinion mining, there are an estimated 33% fake reviews in consumer review sites. Therefore there is need that such types of reviews should be detected and eliminated to provide genuine experience of the business to the users.

The main objective of our project is to build the classifiers using Semi-Supervised machine learning technique. There are various approaches that can be used for semi-supervised machine 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.

2. RELATED WORK

The Opinion mining is a subject which can be analyzed in many ways. Many scholars have done the research in this field, implemented various learning algorithms and have developed several systems that can detect fake reviews, classify spam reviews from non-spam reviews, defined the spammicity of product and so on. Table 1 shown below discusses and compares the various techniques used by scholars in the past to tackle the opinion spamming problem.

First one discusses the learning based approach in general and supervised learning in particular. It uses some behavioral indicators to train its model. Second one from the list discusses the semi-supervised learning based approach which uses active learning technique to classify spam from non-spam reviews. It uses the features such as f-measure, recall, precision [7] etc. to train the model Third from the list explores the collective positive labeled learning technique to train the data model. It also uses the spatial approach such IP tracking to identify the spammers group which deliberately diverts the product sentiment. Forth one deals with k-score analysis to calculate k-values, [8] based on which we could classify spam and non-spam group. It also uses behavioral aspect of connection between the spammers. Lastly, the scholars uses the temporal approach to calculate the spammicity of product, they basically uses the time span analysis at the micro level and cross-site time series anomalies to detect spam behaviors of users. Table 1 also shows that different scholars use different datasets to train and validate their models. So it is bit difficult to compare which technique is better among them in terms of accuracy.

Different scholars implemented and follow different approaches to the subject of sentiment analysis and thus achieve different milestones. We must understand all these methods and techniques to get the basic idea of sentiment analysis domain and understand how problem solving works in this domain.

TABLE I. COMPARATIVE ANALYSIS OF SOME KEY APPROACHES OF SENTIMENT ANALYSIS

A.N.	Dataset Used	Detection Technique Used	Metrics/Features Used
1	Amazon.com	Mixture of Behavioral and Supervised Learning based.	Similarity between reviews, Review frequency, Spamming behaviors
2	Yelp.com	Semi-supervised with active Learning	Precision, Recall, Accuracy, f-measure
3	DiangPing Restro	Positive Unlabeled learning, Spatial Approach (IP tracking), Heterogeneous multitier classification	Reviews :: + ve score -- Fake review - ve score -- Truthful review Reviewers :: + ve score -- Spammer - ve score -- Non- Spammer
4	Mobile01.com	K-score analysis, Behavioral approach	K-Core values, Connection between spammers
5	FourSquare	Temporal approach towards Sentiment analysis, time span study and analyzing pattern anomalies	Micro-level - Time span spam analysis Macro-level - Cross site time series anomalies
6	Amazon.com	Rating deviation, Review Burst, Cosine technique for content similarity.	Precision, Recall, Accuracy, f-measure

TYPES OF SPAMS

Before we deep dive into ocean of sentiment analysis, first let us understand what does the “spam” term means. In a generalized form, spam means any type of message or communication originating from either a person or an organization which is unsolicited or undesired. [4] It usually contains non harmful material such as unwanted advertisement or messages, but sometimes it could be a harmful one containing malware, viruses’ or link to phishing websites etc. Let us understand different types of spams that are exist in real world to get better idea of this term.

A. Email Spam

It a type of spam in which unwanted contents is spread through medium of emails in the form of viruses’, advertisements or messages. Basically Spam emails are nothing but the unwanted emails, most of times they are non-harmful advertisements and messages, but sometimes they contain harmful contents such as malwares or links to phishing websites.

B. Review Spam

It is a type of spam in which the sentiments about particular product or service or individual, are control artificially. Generally people are hired, to post fake reviews in bulk to change the sentiment about the product, either positively or negatively. [4] In this process of opinion spamming the potential buyers are just misdirected or wrongly influenced about particular product or service.

C. Advertisement Spam

These are advertisements of a particular product or service, appear in your browser based on your browsing history. [3] They are posted with the intention of promoting specific product, service or individual.

D. Hyperlink Spam

The addition of external links on the webpage with the intention of promoting particular product or service is a major source of spam in recent times. [4]

E. Citation Spam

These are recently discovered new types of spams, which involves process of illegal citation [4] such as paid citation to improve your scholarly work on internet. These spams generally found in the scholars work.

Cosine Similarity: The cosine similarity is used to calculate the similarity between two non-zero vectors. Here, in this scenario frequency of the words in the sentence is calculated. Considering them as two separate vectors we can easily determine the similarity between two reviews. [12]

It is thus a judgment of orientation and not magnitude: two vectors with the same orientation have a cosine similarity of 1, two vectors oriented at 90° relative to each other have a similarity of 0, and two vectors diametrically opposed have a similarity of -1, independent of their magnitude. The cosine similarity is particularly used in positive space, where the outcome is neatly bounded in [0, 1].

3. PROPOSED MODEL

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.

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 behavioural and linguistic features.

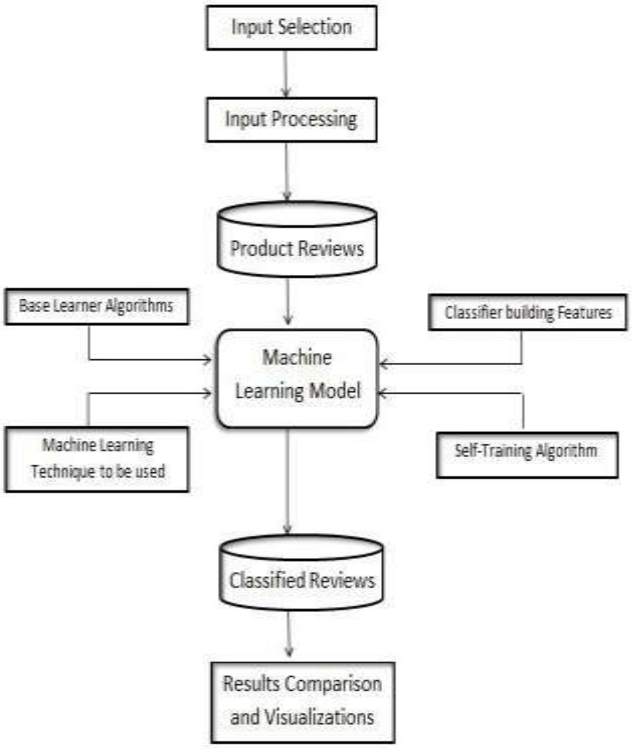

Fig. 1. Proposed model of SSL learning Based FRD

A. 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:

- Restaurant Name
- Average Rating
- User Name
- Review Text
- Rating
- Date of Review
- Classification by Yelp (Recommended / Not Recommended)

B. Preprocessing of Data

We carried out the following steps during preprocessing:

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.

2. Processing 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.

C. Calculating Behavioral Dimensions

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).

TABLE II. LIST OF NOTATIONS USED

Variable - Description	***Description***
a; A; r; r_a =(a,r)	*Author 'a'; set of all authors; 'a' review ;review by author 'a'*
F_{MNR}(a)	*Maximum number of reviews by author 'a'*
Max Rev(a)	*Maximum number of reviews posted in a day by an author 'a'*
f_{rel}	*Length of the review*
f_{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*
Cosine(r_i , r_j)	*Cosine similarity between review i and j*

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.

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

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.

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.

D. 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 was 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.

E. Machine Learning Algorithm

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 and 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.

F. Semi-Supervised Learning

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 nl << nu where nl and nu are the number of labeled and unlabeled data respectively [5].

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 [4].

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.

Algorithm

```
Initialize: L, U, F, T;
 (where, L: Labeled data; U: Unlabeled data;
 F: Underlying classifier; T: Threshold for selection)
Itermax : Number of iterarions; {Pl}Ml=1 : Prior probability;
t  1;
while (U != empty) and (t < Itermax)  do
-           Ht-1  BaseClassifier(L,F);
For each Xi  U    do
-           Assign pseudo-label to X¬i   based on classifier
confidence
-           Sort Newly-labeled examples based on confidence
-           Select a set S of high-confidence predictions according
to ni   P¬i
And threshold T || Selection Step
-           Update U = U – S; and L = L  S;
-           t  t +1
-           Re-Train Ht-1 by new training set L
end while
Output: Generate final hypothesis based on the new training set
```

G. Proposed 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

- Semi-Supervised Vs. Supervised using Logistic Regression:

We aim to implement the base learner as Logistic Regression classifier and use it in the self-training algorithm.

4. EXPERIMENT RESULTS AND DISCUSSION

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. The following graphs show the results of individual base learners vs. the semi-supervised self-training method for varying labeled datasets of yelp reviews.

Result Evaluation:

A. 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.

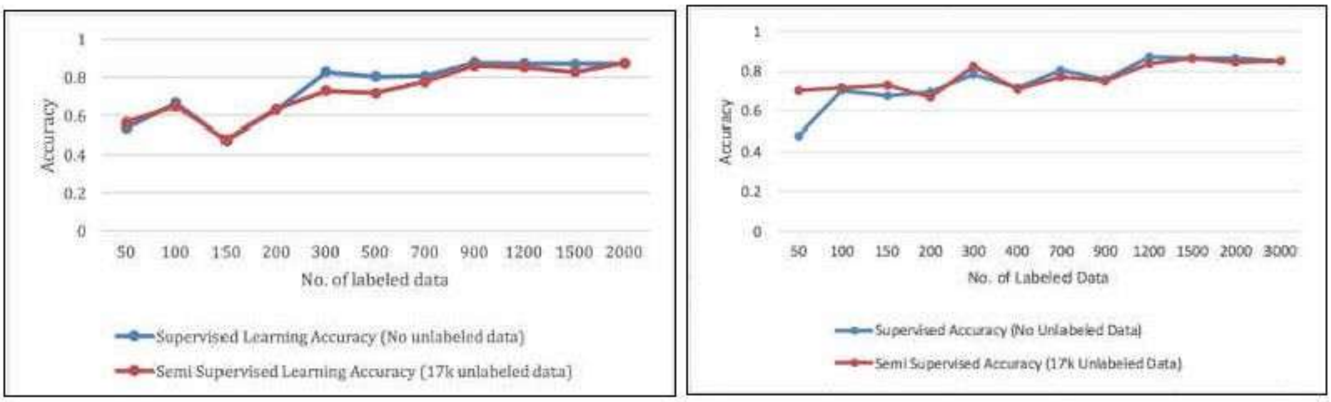

Fig. 2. Semi-Supervised Vs. Supervised using Naïve Bayes (Stratified Sampling on the left and Simple Random Sampling on the right)

B. Critical Evaluation of the Decision Tree 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.

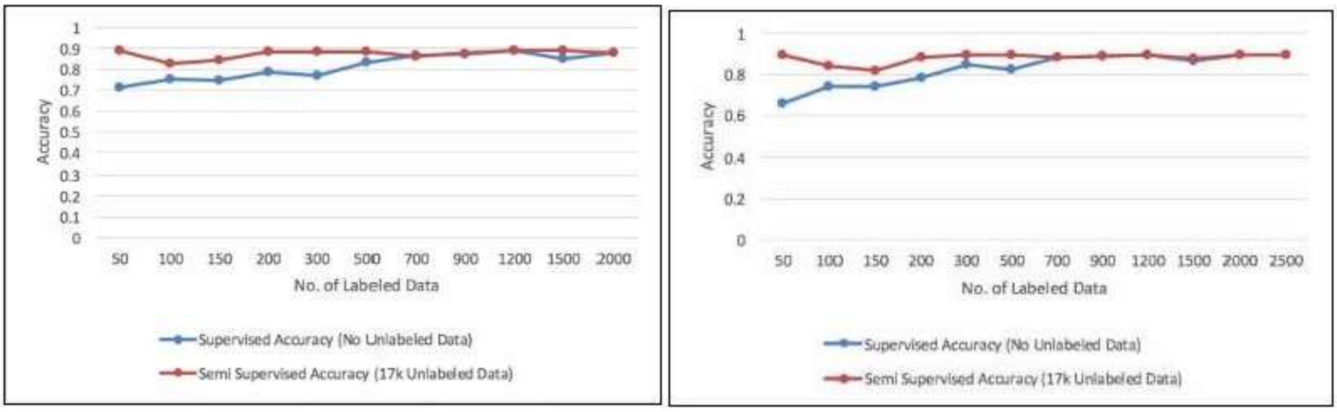

Fig. 3. Semi-Supervised Vs. Supervised using Decision Tree (Stratified Sampling on the left and Simple Random Sampling on the right)

C. Critical Evaluation of the Logistic Regression Experiment

As the size of the labeled data set increases, accuracy of both the models converged to a stable value (Approximately 88%). Thus, Logistic Regression performed well for both the supervised and semi -supervised training model.

For both simple random and stratified sampling using Logistic Regression, 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 of the fact that the self-training algorithm that we're using doesn't work well when the base learner does not produce reliable probability estimates to its predictions.

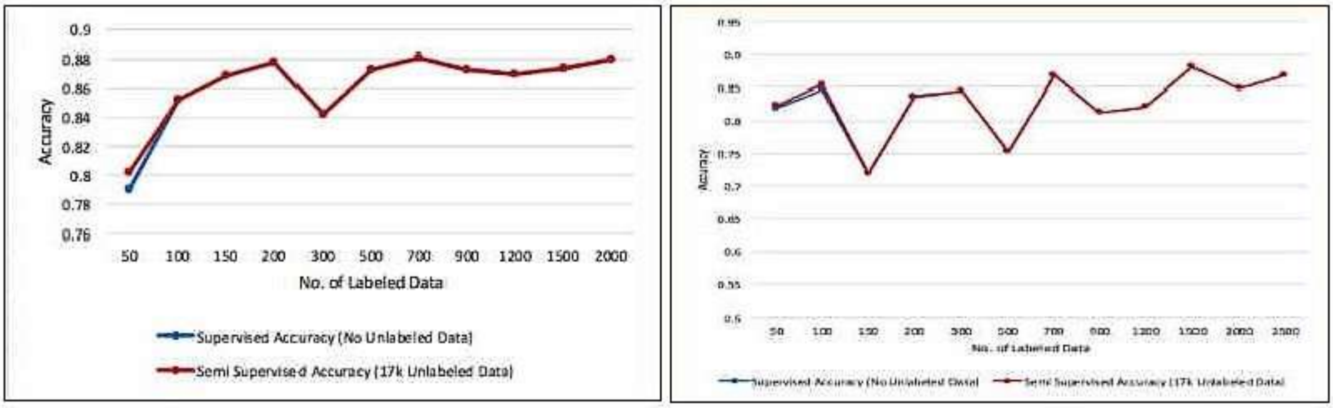

Fig. 4. Semi-Supervised Vs. Supervised using Logistic Regression (Stratified Sampling on the left and Simple Random Sampling on the right)

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] D. Kornack and P. Rakic, "Cell Proliferation without Neurogenesis in Adult Primate Neocortex," Science, vol. 294, Dec. 2001, pp. 2127-2130, doi:10.1126/science.1065467.

[2] A. Mukherjee, Bing Liu, and Natalie Glance, "Spotting fake reviewer groups in consumer reviews," IW3C2,WWW Lyon France, April 2012.

[3] Kolhe N.M., Joshi M.M., Jadhav A.B and Abhang P.D.,"Fake reviewer group's detection system", IOSR-JCE., vol. 16 issue 1., Jan2014, pp.06–09.

[4] Fangtao Li, Minlie Huang, Yi Yang and Xiaoyan Zhu.,"Learning to Identify Review Spam", 22ndInt. joint conf. on AI, China ,2010, pp.2488–2493.

[5] Sindhu C, G.Vadivu, A.Singh and Rahit Patel,"Methods and approaches on spam review detection for sentiment analysis", Int. Journal of pure and applied mathemaatics, vol. 118, No. 22 ,2018, pp.683–690.

[6] Huaxun Deng, Linfeng Zhao, Ning Luo and Yuan Liu,"Semi-supervised learning based fake review detection",IEEE Int. symposiumon parallel and distributed processing with application, 2017

[7] Hang Cui, Vibhu Mittal and Mayur Datar, "Comparative experiments on sentiment classification for online product review" for Americal association for Artificial Intelligence, 2006

[8] Huifeng Tang, Songbo Tan and Xueqi Cheng "A survey on sentiment detection of reviews" Expert systems with applications, China, 2009.

[9] Yahui Xi, Tianjin "Chinise review spam classification using machine learnig method", J Mach. Learn. Res. 3 [march 2003]ICCECT, China, 2012,pp 1289-1305.

[10] Brown, L. D., Hua, H., and Gao, C. "A widget framework for augmented interaction in SCAPE." 2003.

[11] Chirita, P.A., Diederich J. and Nejdl "W.MailRank: Using Ranking for spam detection". CIKM, 2005.

[12] C.L. lai, K.Q. Xu, Raymond Y.K.. IEEE ICEBE. "Towards a language modeling approach for consumer review spam detection".

[13] N. Jindal and B. Liu, "Analyzing and detecting review spam," in Data Mining, 2007. ICDM 2007. Seventh IEEE International conference on, IEEE,2007, pp.547–552.