Summer 2019

Fake Review Detection using Data Mining

Md Forhad Hossain
Missouri State University, Mdforhad08@live.missouristate.edu

FAKE REVIEW DETECTION USING DATA MINING

A Master's Thesis

Presented to

The Graduate College of

Missouri State University

In Partial Fulfillment

Of the Requirements for the Degree

Master's of Science, Computer Science

By

Md Forhad Hossain

August 2019

FAKE REVIEW DETECTION USING DATA MINING

Computer Science

Missouri State University, August 2019

Master's of Science

Md Forhad Hossain

ABSTRACT

Online spam reviews are deceptive evaluations of products and services. They are often carried out as a deliberate manipulation strategy to deceive the readers. Recognizing such reviews is an important but challenging problem. In this work, I try to solve this problem by using different data mining techniques. I explore the strength and weakness of those data mining techniques in detecting fake review. I start with different supervised techniques such as Support Vector Machine (SVM), Multinomial Naive Bayes (MNB), and Multilayer Perceptron. The results attest that all the above mentioned supervised techniques can successfully detect fake review with more than 86% accuracy. Then, I work on a semi-supervised technique which reduces the dimensionality of the input features vector but offers similar performance to existing approaches. I use a combination of topic modeling and SVM for the implementation of the semi-supervised technique. I also compare the results with other approaches that consider all the words of a dataset as input features. I found that topic words are enough as input features to get similar accuracy compared to other approaches where researchers consider all the words as input features. At the end, I propose an unsupervised learning approach named as *Words Basket Analysis* for fake review detection. I utilize five Amazon products review dataset for an experiment and report the performance of the proposed on these datasets.

KEYWORDS: data mining; deceptive reviews; topic modeling; SVM; opinion spam; words basket analysis

FAKE REVIEW DETECTION USING DATA MINING

By

Md Forhad Hossain

A Master's Thesis
Submitted to The Graduate College
Of Missouri State University
In Partial Fulfillment of the Requirements
For the Degree of Master's of Science, Computer Science

August 2019

Approved:

Jamil M. Saquer, Ph.D., Thesis Committee Chair

Razib Iqbal, Ph.D., Committee Member

Lloyd A. Smith, Ph.D., Committee Member

Julie Masterson, Ph.D., Dean of the Graduate College

ACKNOWLEDGEMENTS

First, I would like to thank Dr. Jamil M. Saquer for his continuous help throughout the research. His guidance helped me a lot to make good progress in my research. I enjoyed and learned a lot working with him. He also helped me a lot for my teaching assistant (TA) duties. He is very well organized, which made my TA duties easier. I would also like to thank Dr. Razib Iqbal for helping me in my research. I liked his always-ready attitude to help. He also helped me a lot outside of academic work. I would also like to thank Dr. Lloyd A. Smith for helping with all the academic works and GA duties.

Lastly, I would like to thank my family for their continuous support from Bangladesh. I dedicate this thesis to my mother (Fatema), my elder brother (Faruk), my younger sister (Jemmy), my younger brother (Tanvir) and my uncle (Amjadh).

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1 INTRODUCTION

E-commerce is growing at an unprecedented rate all over the globe. With its growth, the impact of online reviews is increasing day by day. Reviews can influence people's purchasing decisions. Nowadays, reading product reviews before buying the product has become a habit, especially for potential customers. Customers post reviews about a product they purchase which may be positive or negative. Such reviews provide valuable feedback on these products, which may further be used by potential customers to find the opinions of existing users before deciding to purchase a product. If customers want to buy a product, they usually read reviews from some customers about the current product. If the reviews are mostly positive, there is a big chance to buy the product. Otherwise, if the reviews are mostly negative, customers tend to buy other products.

While online reviews can be helpful, blind trust of these reviews is dangerous for both the seller and buyer. Most customers read online reviews before placing any online order. However, the reviews may be deceptive for extra profit or gain, thus any purchasing decision based on online reviews must be made carefully. To sell their products, companies often pursue customers to give desired reviews. There is a growing incentive for businesses to solicit and manufacture deceptive reviews, a.k.a. opinion spam- fictitious reviews that have been deliberately written to sound authentic and deceive the reader [1]. For example, Ott [2] has estimated that between 1% and 6% of positive hotel reviews appear to be deceptive, suggesting that some hotels may be posting fake positive reviews in order to hype their own offerings.

Spam detection has been studied in many areas. Web spam and e-mail spam are the two most widely studied types of spam. Opinion spam is very different from those two. Unlike other forms of spam, it is almost impossible, to recognize fake opinions by manually reading them. For example, one can write a truthful review of a bad hotel and post it as a fake review for a good hotel. Fake reviews are especially damaging for small businesses. Even a single bad fake review can cause significant damage to a small business. A couple of examples of truthful and deceptive

reviews about a restaurant named Affinia in Chicago city are given below. We request the reader to read those reviews carefully and label them. You can check your answer later.

Review 1. I was completely blown away by this hotel. It was magnificent. I got a great deal and I am so happy that I stayed here. Before arriving I was nervous as I had read a few bad reviews about the impact the renovation was having on peoples stay, for example very noisy. However, whilst the renovation was still going on and the gym was not open nor the restaurant, it made no difference to me. My room was huge, bathroom was spacious with excellent water pressure, bed was perfect and the view was amazing. Hotel is so close to the great shops of Magnificent Mile, plus a comfortable walking distance to Hancock tower and Millennium Park.

Review 2. My husband and I arrived for a 3 night stay for our 10th wedding anniversary. We had booked an Executive Guest room, upon arrival we were informed that they would be upgrading us to a beautiful Junior Suite. This was just a wonderful unexpected plus to our beautifully planned weekend. The front desk manager was professional and made us feel warmly welcomed. The Chicago Affinia was just a gorgeous hotel, friendly staff, lovely food and great atmosphere. Not the mention the feather pillows and bedding that was just fantastic. Also we were allowed to bring out beloved Shi-Tzu and he experienced the Jet Set Pets stay. The grooming was perfect, the daycare service we felt completely comfortable with. This was a beautiful weekend, thank you Affinia Hotels! We would visit this hotel again!

Review 3. There were many positives when staying in this hotel at the north end of the Magnificant Mile. The rooms were spacious and beautifully appointed with great attention to detail. The quality of service was excellent, the staff professional, and the location was perfect. There are great views of Chicago from many of the rooms. The staff seemed to care about the quality of my stay from the moment I was greeted upon arrival to the moment I got in my taxi to leave. There is a nice rooftop bar/restaurant perfect for a lite meal or a refreshing cocktail with great views of the city. There really weren't any negatives and I would recommend this boutique hotel to anyone staying in the Chicago area.

Review 4. Stayed at this hotel with 3 friends or 4 nights. The hotel was clean and tidy,

throughout. The Concierge Christopher was excellent and helped as with all our needs, gave us discount vouchers etc. Hotel was in excellent position. 3 blocks from John Hancok Building and more importantly The Cheesecake Factory , Bloomingdales 3 blocks away and the Water Tower Shopping Centre 2 blocks away. We went to the huge I Max cinema which is about ten minutes away. Cant wait to go back to Chicago and The Affinia

Review 1 and Review 4 are truthful whereas Review 2 and Review 3 are deceptive. Compare your labeled answer with the actual result. Now you will have an idea of how difficult it is to detect a fake review. Its almost impossible to label a review correctly simply reading the review.

Besides, day by day customers in the USA are becoming concerned about fake reviews. The Google trend chart for ”fake review” (Figure 1.1) clearly tells us about customers concern regarding counterfeit reviews. In Figure 1.1, numbers represent search interest relative to the highest point on the chart. A value of 100 is the peak popularity for the term. A value of 50 means that the term is half as popular. A score of 0 means there was not enough data for this term.

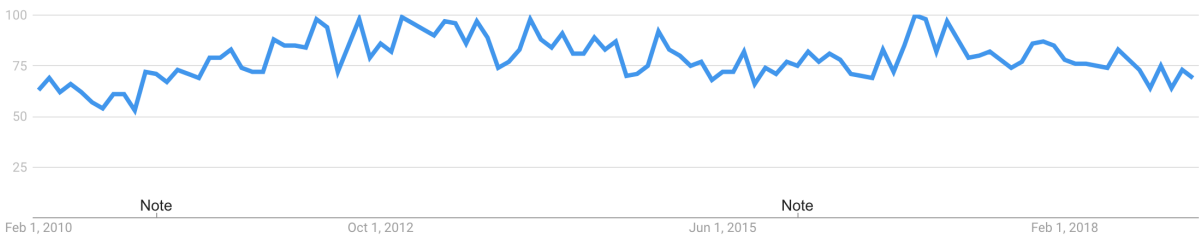

Figure 1.1: The Google Trend result of the search query ”Fake Review” in the USA

Primarily, there are three types of spam reviews [3]:

1) Fake review: These reviews are written with a motive to promote or demote a product. In most of the cases reviewers do not have first-hand experience of using the products or services.

2) Review about brands only: Here reviews are not about a specific product or service instead it is about the brand or manufacturer of the product. For example, a review about Samsung smartphone that says, ”I hate Samsung phone, I never buy any of their smartphones.”

3) Non-reviews: These reviews are advertisements and irrelevant content without any proper opinion.

There are two types of fake reviewers: professional fake reviewers and nonprofessional fake reviewers.

Professional fake reviewers: A professional fake reviewer usually writes a large number of reviews. They might work as a freelancer or work for a company to write fake reviews. They get paid for their work. As they write a large number of reviews, their linguistic and behavioral patterns can be easily identified by different data mining algorithms. But before they are caught they might have already done a significant amount of damage. To make matters worse, when they are caught for spamming reviews they leave their account, and create a new account and start spamming again.

Nonprofessional fake reviewers: They write a small number of fake reviews. These people write fake reviews to foster themselves or their friends and families. They usually do not get paid for their work. As they write a small number of fake reviews, their linguistic and behavioral patterns are hard to identify.

Spamming can occur in two ways: individual spamming or group spamming.

Individual spamming: An individual spammer does not work with anyone. For example, a seller of a product may write fake reviews for himself to promote his business or product.

Group spamming: A group of spammers work together to promote or demote a product or service. It usually carried out by professional spammers. Group spamming is more damaging than individual spamming.

There are three main types of data that can be used for spam review detection:

Review Content: We can extract linguistic features such as word and parts of speech (POS), n-grams and other syntactic, semantic, and stylistic clues for deceptions and lies.

Meta-data about each review: The data such as the star rating given to each review, user id of reviewer, review-id, the time/date when the review was posted, and the number of helpfulness votes.

Web usage data: This data include the sequence of clicks, the time when each click is made, how much time a user stays on a page, the time taken to write a review and so on. Such data are also called side information.

Product Information: Information about the entity being reviewed, for example, product brand, model, type/category, and description.

Sales Information: This mainly includes business-related information such as the sales volume and the sales rank of a product in each period of time.

Generally, lying/deception communications are characterized by the use of fewer first-person pronouns, more negative emotion words, fewer exclusive words, and more motion/action words [3]. Several researchers have hypothesized that liars often try to avoid statements of ownership either to dissociate themselves from their words or due to the lack of personal experiences [3]. Fake reviewers do not give any specific opinion about any specific features of products or services as they lack the first-hand experience. So they use more general opinion words such as great, wonderful, and so on in their given reviews.

Detecting fake review using supervised learning is applicable. It is possible to use supervised learning to detect fake reviews, because fake review detection can be formulated as a classification problem with two classes, fake and non-fake. However, the key difficulty is that it is almost impossible to recognize fake reviews reliably by manually reading them because a spammer can carefully craft a fake review that is just like any genuine review. For this reason, there is no reliable fake review and non-fake review data set available to train a machine learning model to recognize fake reviews. Because there is no labeled data for learning, Jindal and Liu [4] exploited duplicate reviews in their study on Amazon reviews. Researchers employed three sets of features for learning: Review-centric features, Reviewer-centric features, and Product-centric features.

Review-centric features: These features are all about each review. These features include length of the review, number of useful votes, review rating, percentage of positive and negative sentiment words in the review, cosine similarity of the review and the product description, percentage of brand name mentions, etc.

Reviewer-centric features: These features are about each reviewer. These features include review frequency of the reviewer, average rating given by the reviewer, total number of reviews posted, etc.

Product-centric features: These features are about each product. These features include the average rating of the product, price of the product, brand name of the product, etc.

Li et al. [5] manually labeled a set of fake and non-fake reviews by reading the reviews and the comments. We have described in more details about their labeling process in the next chapter. Ott et al. [6] used Amazon Mechanical Turk to crowdsource fake hotel reviews of twenty hotels. Their work reported 89.6% of accuracy using only word bigram features under the balanced class distribution. Feng et al. [7] used some deep syntax rule-based features to boot the accuracy to 91.2%.

There are two types of features that can be used in classification: linguistic features and behavioral features [3]. Linguistic features are about the review text content, while behavioral features are about behaviors of reviewers and their reviews. Behavioral features combined with bigram gave the highest accuracy with Yelp data (only contains positive reviews about popular Chicago hotels and restaurants). Wang et al. [8] proposed a graph-based module for detecting spam where reviews describe purchase experiences and evaluations of stores. The spamming behaviors are as follows: targeting products, targeting groups, general rating deviation, and early rating deviation [3].

Targeting products: Spammer usually put their efforts toward promoting or demoting a few target products. They closely monitor those products and mitigate the rating by giving fake reviews.

Targeting groups: It defines the pattern of spammers manipulating ratings of a set of products sharing some attributes within a short span of time.

General rating deviation: A real reviewer usually gives a rating to a product that is almost similar to other raters of the same product. However, a spammer tries to promote or demote a product by giving a rating that typically deviates from those of other reviews.

Early rating deviation: Early deviation catches the behavior of a spammer by giving a fake review shortly after product launch. Such reviews are likely to draw attention from other reviewers which let spammers to alter the views of consequent reviewers.

In our research, we mainly focus on review-centric features of each review. We apply supervised, semi-supervised and unsupervised methods for detecting fake reviews. We do not utilize any individual or group behavior features in our proposed method for fake review detection. We use a publicly available hotel reviews dataset and five different Amazon products reviews for our research.

2 RELATED WORK

Most of the research work on spam review detection falls into two categories. One group of researchers focus on only the content of the reviews. On the other hand, other groups of researchers concentrate on reviewers behavior instead of review content. But a combination of both approaches gives the best result.

In [4], Jindal, et al. claimed they are the first to attempt to study review spam and spam detection. They collected 2.14 million reviews from Amazon for their research work. They found a large number of duplicate and near-duplicate reviews written by the same reviewers on different products or by different reviewers on the same products or different products. They proposed to perform spam detection based on duplicate finding and classification. They used logistic regression to learn a predictive model. Using 10-fold cross-validation on the data they got average area under the ROC curve (AUC) value of 78%.

In [6], Ott, et al. showed that psychological studies of deception and genre identification are both out-performed at statistically significant levels by n-gram based text categorization techniques. Notably, a combined classifier with both n-gram and psychological deception features achieves nearly 90% accuracy.

In [9], Ott, et al. worked on negative deceptive opinion spam which usually are reviews that aim at degrading other company's reputations. They found that standard n-gram text categorization techniques can detect negative deceptive opinion spam with performance far surpassing that of human judges.

In [10], Sandulescu, et al. used one time reviewers such as a reviewer who leaves only one review. They exploited the singleton reviewers review. They tackled the problem of detecting fake reviews written by the same person using multiple names, posting each review under a different name. They propose two methods to detect similar reviews and show the results generally outperform the vectorial similarity measures used in previous work. Their proposed methods

are the semantic similarity between words to the review level and based on topic modeling and exploit the similarity of the reviews topic distributions using two models: bag-of-words (a simplifying representation used in natural language processing and information retrieval) and bag-of-opinion (a simplifying representation used in natural language processing and opinion mining) phrases.

In [5], Li, et al. manually labeled nearly 6000 reviews. They collected a dataset from the Epinions website. They employed ten college students for tagging all the reviews. Students were first instructed to read books and articles about how spam review looks like then they were asked to label those reviews.They first used supervised learning algorithm and analyze the effectiveness of different features in review spam identification. They also used a two-view semi-supervised methodology to exploit a large amount of unlabeled data. The experiment results show that two-view co-training algorithms can achieve better results than the single-view algorithm.

In [11], Luca, et al. worked on restaurant reviews that are identified by Yelp's filtering algorithm as suspicious, or fake. They found that nearly one out of five reviews is marked as fake by Yelp's Algorithm. These reviews tend to be more extreme than other reviews and are written by reviewers with less established reputations. Moreover, their finding suggests that economic incentives factor heavily into the decision to commit fraud. Organizations are more likely to game the system when they are facing increased competition and when they have poor or less established reputations.

In [12], Wahyuni, et al. aimed to detect fake reviews for a product by using the text and rating property from a review. Their proposed system measures the honesty value of a review, the trustiness value of the reviewer and the reliability value of a product.

In [13], Jindal, et al. deal with identifying unusual review patterns which can represent suspicious behaviors of reviewers. They formulate the problem as finding unexpected rules. They analyzed an Amazon.com review dataset and found many unexpected rules and rule groups which indicate spam activities.

In [14], Lim, et al. recognize spammers based on behaviors of reviewers that deviate from

usual practice. These reviewers are highly suspicious of review manipulation. Their research suggests that one should focus on detecting spammers based on their spamming behaviors, instead of identifying spam reviews. Their proposed review spammer detecting approach is user-centric, and user behavior-driven. They claimed their proposed methods generally outperform the baseline method based on helpfulness votes.

In [15], Mukherjee et al. dove down to Yelp's secret filtering algorithm.They put a few existing research methods to the test and evaluated performance on the real-life Yelp data. They found the behavioral features perform very well, but the linguistic features are not as effective. Their analysis and experimental results shows that Yelp's filtering is reasonable and its filtering algorithm seems to be correlated with abnormal spamming behaviors.

In [16], Li, et al. claimed they are the first one to present a large-scale analysis of restaurant reviews. They were able to collect a large amount of data from Dianping which is a Chinese group buying website for locally found food delivery services, consumer products and retail services. Dianping helped them to get user reviews about restaurants and, users IP addresses and profiles. They used a method called Positive-Unlabeled Learning. They used temporal and spatial features at various levels (reviews, users, IPs) for supervised opinion spam detection.

In [17], Xie, et al. developed a model for singleton spam review detection problem based on the observation that the arrival pattern of singleton review tends to be bursty and temporally correlated to the rating.

In [18], Li, et al. worked on detecting spamming network using reviewer posting frequency within short periods of times and also considered other users posting frequency within that short period of time for the same products. They primarily tried to find out individual spammers and spammer groups.

In [19], KC, et al. worked on the temporal dynamics of opinion spamming. They looked to find out if there are any specific spamming policies that spammers employ. They used a large set of reviews from Yelp restaurants and its filtered reviews to characterize the way opinion spamming operates in a commercial setting. Using time-series analysis, they found that there exist

three dominant spamming policies: early, mid and late across the various restaurant. Their analysis showed that the deception rating time-series for each restaurant had statistically significant correlations with the dynamics of truthful rating time-series indicating that spam injection may potentially be coordinated by the restaurants/spammers to counter the effect of unfavorable rating over time.

In [20], Shebuti and Akoglu proposed a framework named Speagle that exploits both relational data (user-review-product graph) and metadata (behavioral and text data) collectively to detect suspicious users and reviews, as well as products targeted by spam. Their main contribution is to employ a review-network-based classification task which accepts prior knowledge on the class distribution of the nodes, estimated from metadata. Their proposed framework works in an unsupervised fashion, but can easily leverage labels.

In [21], Hooi, et al. used a Bayesian Model approach to detect spam reviews. They considered two parameters for fraud review detection. One, Review in short time bursts/periods and another one is finding users who rate product very differently than others.

In [22], Li, et al. used Collective positive-unlabeled (PU) learning for fake review detection. They proposed a supervised algorithm called Multi-typed Heterogeneous Collective Classification (MHCC) for the heterogeneous network of reviews, users and IPs. Then they extended it to Collective Positive and Unlabeled learning (CPU). Their results show that their proposed models can remarkably improve the F1 scores of strong baselines in both PU and non-PU learning settings.

While the vast majority of existing methods focused on either review text or behavioral analysis for detecting a spam review, however in [23], Akoglu et al. used a graph-based network effect among reviewers and products. Their proposed model entirely operates in an unsupervised fashion and is linearly scalable. It consists of two complementary steps; scoring users and reviews for fraud detection, and grouping for visualization and sense-making. They found reviewers, reviews, and products are more deeply encapsulate structure signals. They have created a bipartite network among products, users and reviews for spam review detection.

In [24], Jindal and Liu made the first attempt to investigate opinion spam in reviews and proposed some novel techniques to study spam detection. They used manual labeling for review on brands only and non-reviews. But for untruthful opinions, they used a large number of duplicate and near-duplicate reviews to build a spam detection model.

In [25], Chauhan et al. incorporated sentiment analysis of reviews techniques into the spam review detection. They used sentiment analysis using their in house-dictionary and compared sentiment analysis result of a product with the given review by customers. If both results rating difference is higher than a certain level(e.g. 0.5) they level it as a Spam.

In [26], Li, et al. proposed a generative LDA-based topic modeling approach for fake review detection. Their approach is a variation of Latent Dirichlet Allocation (LDA) and aims to detect subtle differences between the topic-word distributions of deceptive reviews vs truthful ones [27].

3 SUPERVISED LEARNING APPROACH

In this section, we will talk about the different approaches we take to find fake reviews. We use both supervised and unsupervised methods. We explore different data mining algorithm's performance in fake review detection and compare their results. We also discuss the dataset collection procedure.

3.1 Data Collection

We use two types of datasets in our research. One dataset is labeled and another one is unlabeled. The biggest problem we faced in our research is finding a labeled dataset. We found only one publicly available labeled dataset. We also collected different Amazon products reviews which are unlabeled. In this section, we will discuss in details how the labeled and unlabeled datasets were collected.

To conduct most of our experiments, we used Ott et al.'s publicly available dataset of opinion spam [6]. The dataset contains 800 positive reviews (400 truthful and 400 deceptive) and 800 negative reviews (400 truthful and 400 deceptive) of 20 popular Chicago hotels. To collect deceptive reviews, the authors used Amazon Mechanical Turk (AMT) which is a popular crowdsourcing service. AMT has a large number of well-educated anonymous online workers (known as Turkers). Turkers help to make large-scale data annotation with a small amount of money. The authors created a pool of 400 Human-Intelligence Tasks (HITs) and allocated them evenly across 20 chosen hotels. To ensure that unique authors write opinions, they allowed only a single submission per Turker. They took into consideration that Turkers live in the United States and have an approval rating of at least 90%.

In order to collect truthful reviews, the authors mined 6977 reviews from the 20 most popular Chicago hotels on TripAdvisor. They filtered reviews depending on the following: 1) non-5-star reviews, 2) non-English reviews, 3) reviews with fewer than 150 characters and 4) reviews written by first-time authors —new users who have not previously posted a review on TripAdvi-

sor.

We use the phrase *positive reviews dataset* to refer to the part of the dataset that contains truthful and deceptive reviews that promote a product. Likewise, we use the phrase *negative reviews dataset* to refer to the part of the dataset that contains truthful and deceptive reviews that demote a product.

We also collected a review dataset of Amazon products. The products include a set of headphones, iPhone, blending machine, power bank, and a book. To select these products, we mainly focused on the following two criteria: 1) product's popularity, and 2) availability of numerous number of reviews (e.g. 1k reviews). We used web scraping to collect reviews from Amazon website. For web scraping, we used Python's Requests and BeautifulSoup4 packages. Requests package is used for performing HTTP requests and BeautifulSoup4 is used for handling all of the HTML processing. We collect the content from the specific URL by making an HTTP GET request. If the content-type of the response is some kind of HTML/XML, it returns text content, otherwise, it returns None. Returned content is raw HTML. The BeautifulSoup4 constructor parses raw HTM strings and produces an object that contains the HTML document structure. The object includes a slew of methods to select, view, and manipulate the document object model (DOM) nodes and text content. From all the DOM, we only take the DOM that contains product review related information. We collect the following information from each review: rating, title, date, verified purchase, body and helpful votes. Each Amazon page contains ten reviews. Therefore, each time we send a URL request, it returns ten reviews. Amazon blocks the IP address if it receives continues requests from a single IP address. To avoid continually sending URL requests, we decided to send a request in a random interval between 5 to 10 seconds. Following the above procedure, we collected 1000 reviews from one blending machine, 5000 reviews from one headphone, 2259 reviews from one iPhone 6S, 1700 reviews from one book and 2610 reviews from one power bank.

3.2 Algorithm Selection

One group of researchers focused on features associated with the behavior of the reviewer for fake review detection. Another group of researchers solely focused on review text content for fake review detection. In our research, we mainly concentrate on review text content due to the available dataset. There are few approaches such as bags of words, psycholinguistics analysis, and n-gram based that are used for detecting fake reviews. However, standard n-gram based text categorization techniques have been shown to be effective at detecting deception in the text [1, 6, 24], and the text classification Support Vector Machine (SVM) comparatively performs better than other data mining algorithms[15]. For this reasons, we decided to use SVM with unigram and bigram term frequency for getting better performance in detecting spam reviews. In addition to SVM, we used Multinomial Naive Bayes (MNB) and Multi-Layer perceptron (MLP) for detecting spam reviews since they also perform well for text classification.

3.3 Different Algorithms' Performance

In this section, we use Support Vector Machines (SVM) with stochastic gradient descent (SGD) learning, Multinomial Naive Bayes (MNB), Multilayer Perceptron with one hidden layer (MLP1) and two hidden layers (MLP2) for classification. We report the deception review detection performance of the above classifiers. We evaluate the performance using Ott et al.'s [6] positive and negative deceptive review dataset.

The outline of our approach for detecting spam reviews using SVM is shown in Algorithm 1. We use the Scikit-learn package from Python in our implementation. We create a documentList from the input dataset in lines 1. We use 5-fold cross-validation for evaluating the model's performance. For 5-fold cross-validation, the dataset is divided into 5 parts, where 4 parts are used for training the model and one part is used for testing the model's performance. This process is repeated five times as shown by the *for* loop in line 2. Therefore, each tuple in the dataset is used once for testing and four times for training. In line 3, we split the documentsList into training and testing data. In line 4, we tokenize the training data and count the occurrence of

each token in the documents. We discard all the stopwords. Normalization and weighting are also performed in line 4 to diminish the importance of tokens that occur in the majority of the documents. In line 5, we apply the TFIDF Transformer to the output from the previous line to give similar priority to long and short documents and perform weight downscaling. We build the classifier/model using linear SVM with stochastic gradient descent (SGD) learning in line 6 and test it in line 7. We also use "Held Out" for evaluating the generated model's performance. In "Held Out" process, we train the model with positive reviews and test with negative reviews and vice versa. We follow a similar procedure for building a classifier with MNB, MLP1, and MLP2.

Algorithm 1 Algorithm for detecting spam reviews using SVM

```
1: documentsList= input dataset
2: for i = 1 to 5 do
3:     training_data, testing_data = Split documentsList
4:     tokenize = CountVectorizer(training_data).fit_transform(training_data)
5:     tfidf = TfidfTransformer().fit_transform(tokenize)
6:     classifier = SGDClassifier().fit(tfidf)
7:     prediction = classifier.predict(testing_data)
8: end for
```

Results appear in Tables 3.1 - 3.4. If we look at the performance of the generated model in Table 3.1, we find that for positive reviews, the accuracy of 5-fold cross-validation is nearly 89%. But when we follow the "Held Out" procedure to evaluate the performance of the generated model, its accuracy dropped to only 75%. The "Held Out" testing procedure gives better performance with negative reviews, however, and accuracy is nearly 81% and almost 90% for 5-fold cross-validation. This implies that people use different sets of words for a negative deceptive review than for a positive deceptive review. Because of this, we need to handle a negative deceptive review differently. However, 5-fold cross-validation with a combined dataset of positive and negative reviews comparatively performs better than 5-fold cross-validation with only positive or negative reviews. We get the highest precision (almost 93%) in 5-fold cross-validation with negative reviews and the lowest precision (nearly 70%) in "Held Out" with positive reviews.

In Table 3.2, we outline the performance of MNB classifier. With positive dataset following the cross-validation procedure it gives 86.56% accuracy, but following the "Held Out" procedure it gives only 75.00% accuracy. However, with negative dataset following the cross-validation it offers 86.56% accuracy and following the "Held Out" it offers 85.50% accuracy. We get the highest precision (almost 95%) in cross-validation both with positive and negative reviews and the lowest precision (nearly 75%) in "Held Out" with positive reviews.

In text classification, Artificial Neural Networks show a promising result [28, 29]. For this reason, we also tried to check ANN performance with our dataset. We used Multi-Layer Perceptron (MLP) for classification. We build our model with Scikit-learn MLPClassifier function which trains using backpropagation. We use one and two hidden layers where the number of nodes in each hidden layer ranges from 2 to 50. This is because in our experiment, we observed that we get the best performance when we use the number of nodes in a hidden layer between 2 to 50. After this, we determine the best number of nodes that gives the best performance. Using the negative reviews dataset when we use one hidden layer, we obtain the best accuracy (86.5%) with 26 hidden nodes. Similarly, using negative and combined reviews dataset, we get the best accuracy with one hidden layers with 13 and 35 hidden nodes numbers (87.87% and 88.5%, respectively). We use cross-validation for this testing. In table 3.3, we show multilayer perceptron classifier's full details performance with one hidden layer. We also test with "Held Out". For positive and negative reviews, we obtain the best accuracy (77.25% and 84%), when each hidden layer contains respectively 47 and 44 nodes. The highest precision we get (93.03%) in cross-validation with negative reviews and the lowest precision we get (73.39%) in "Held Out" with positive reviews.

Similarly, we also check MLP classifier's performance with two hidden layers which is shown in Table 3.4. Using negative, positive and combined dataset, we get the best accuracy respectively 86%, 88.37% and 88.5%, when each hidden layer contains respectively 15, 27 and 41 nodes. When we test "Held Out" with positive review dataset, two hidden layers and the number of nodes in each hidden layer is 12. This gives us 77.37%, the best result. We receive the highest

precision (almost 94%) in cross-validation with negative reviews and the lowest precision (nearly 75%) in "Held Out" with positive reviews.

Table 3.1: Automated SGD Classifier's performance with different train and test sets, including (A)ccuracy, (P)recision, (R)ecall and (F)1-score.

			TRUTHFUL			DECEPTIVE		
Training	Testing	Accuracy	P	R	F-1	P	R	F-1
Positive (800 reviews)	Cross-Validation	89.12	90.37	87.44	88.81	87.94	90.89	89.32
	Held-Out	75.50	**70.00**	89.00	78.00	85.00	62.00	72.00
Negative (800 reviews)	Cross-Validation	88.50	**92.40**	83.68	87.79	85.15	93.39	89.04
	Held-Out	81.62	76.00	92.00	83.00	90.00	71.00	79.00
Combined	Cross-Validation	89.37	90.55	88.07	89.23	88.39	90.75	89.50

Table 3.2: Automated MNB classifier's performance with different train and test sets, including (A)ccuracy, (P)recision, (R)ecall and (F)1-score.

			TRUTHFUL			DECEPTIVE		
Training	Testing	Accuracy	P	R	F-1	P	R	F-1
Positive (800 reviews)	Cross-Validation	86.56	**94.82**	77.47	85.18	80.99	95.76	87.70
	Held-Out	75.00	**70.00**	87.00	78.00	83.00	63.00	72.00
Negative (800 reviews)	Cross-Validation	86.56	**94.82**	77.47	85.18	80.99	95.76	87.70
	Held-Out	85.50	85.00	83.00	84.00	84.00	85.00	85.00
Combined	Cross-Validation	86.56	94.82	77.47	85.18	80.99	95.76	87.70

Table 3.3: Automated MLP classifier's performance using one hidden layer with different train and test sets, including (A)ccuracy, (P)recision, (R)ecall and (F)1-score.

			TRUTHFUL			DECEPTIVE		
Training	Testing	Accuracy	P	R	F-1	P	R	F-1
Positive (800 reviews)	Cross-Validation	87.87	89.66	85.36	87.42	86.16	90.36	88.17
	Held-Out	77.25	**73.39**	85.50	78.98	82.63	69.00	75.20
Negative (800 reviews)	Cross-Validation	86.50	**93.03**	78.74	85.13	81.77	94.46	87.53
	Held-Out	84.00	80.49	89.75	84.86	88.41	78.25	83.02
Combined	Cross-Validation	88.50	92.53	83.84	87.92	85.23	93.23	89.01

Table 3.4: Automated MLP classifier's performance using two hidden layer with different train and test sets, including (A)ccuracy, (P)recision, (R)ecall and (F)1-score.

			TRUTHFUL			DECEPTIVE		
Training	Testing	Accuracy	P	R	F-1	P	R	F-1
Positive (800 reviews)	Cross-Validation	88.37	89.65	86.73	88.09	87.14	90.18	88.55
	Held-Out	77.37	**74.49**	83.25	78.63	81.01	71.50	75.96
Negative (800 reviews)	Cross-Validation	86.00	**93.81**	76.99	84.38	80.69	95.18	87.20
	Held-Out	84.25	80.71	90.00	85.10	88.70	78.50	83.28
Combined	Cross-Validation	88.50	91.79	84.64	88.04	85.76	92.23	88.90

3.4 Visualization of POS Frequency

In this section, we try to visualize in Figure 3.1 and 3.2 different parts of speech (POS) present in positive, negative, truthful and deceptive reviews.

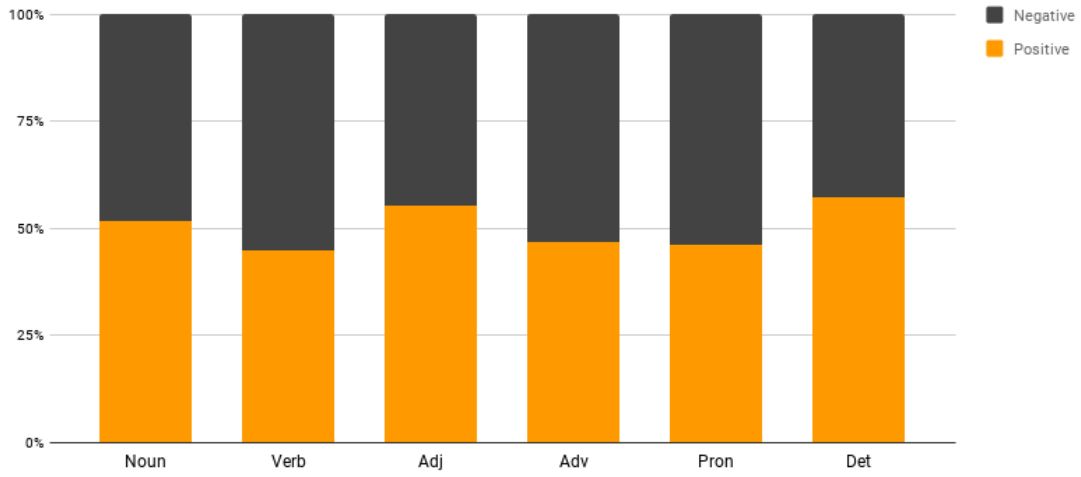

Figure 3.1: Parts of Speech(POS) frequency distribution over positive and negative review.

3.5 Working Procedure and Results

We build two groups of datasets, positive and negative, and another truthful and deceptive review. Using all of the positive and negative reviews, we determine the most frequently used POS. To do that, we use Python's NLTK package. First, we tokenize all of the reviews and remove stopwords. Then we use the pos_tag for tagging all filtered tokens. For tagging, we use

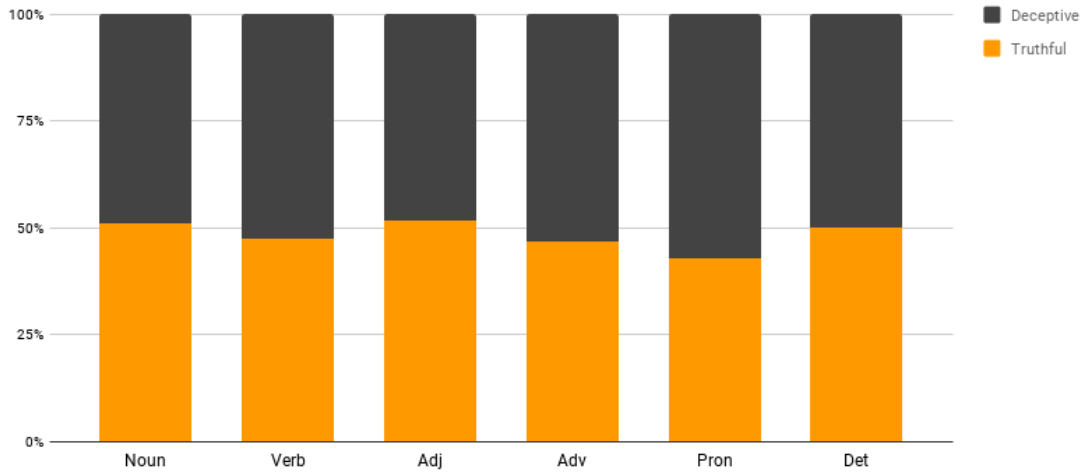

Figure 3.2: Parts of Speech(POS) frequency distribution over truthful and deceptive review.

universal tagset. Using FreqDist function of NLTK we find out the most frequently used POS. We only consider the following parts of speech: noun, verb, adj, adv, pron, and det for our experiment. We do not consider other POS with an overall percentage of presence less than 3%.

We observe that people use nouns more frequently than other parts of speech. From visualizing Figures 3.1 and 3.2, we can say that for positive and truthful reviews people use nouns and adjectives more than other POS. For negative and deceptive reviews, they use verb, adv, pron and det more frequently than other POS. For positive reviews, people use nearly 3% more nouns than for negative reviews. A similar result is found with adjectives. Verbs were used 6% less in positive reviews than in negative reviews. Between truthful and deceptive reviews, maximum difference of the percentage of different POS is only 2%. From the above results, we can say that it is hard to label a review using only POS frequency.

4 TOPIC MODELING BASED APPROACH

Topic modeling is an approach for discovering topics from a large corpus of text documents. The most common output of a topic model is a set of word clusters and a topic distribution for each document. Each word cluster is called a topic and is a probability distribution over words in the corpus. Topics are aspects that refer to the attributes and components of an entity. Aspects are the most important parts to get inside a document. From looking at aspects, we can deduce what is the document about. We assume that aspects of deceptive reviews will be similar. Same goes for truthful reviews. For example, in the sentence "The picture and xsound quality of Samsung-HDTV are great.", picture and sound are aspects of the Samsung-HDTV. As per [3], the advantage of topic modeling is that it can automatically extract aspects and put them into separate groups. For example, it can extract and group organization, cleanliness, and comfort under one topic in a hotel review dataset.

In this section, we present an approach that uses Topic Modeling and Support Vector Machines (SVM) to detect both deceptive positive and deceptive negative reviews. Our approach uses only the topic words that are generated by topic modeling, compared to the existing approaches, e.g. [6, 9], that use all the words in the dataset.

While earlier works, e.g. [6, 18, 30], explored different characteristics of reviews and reviewers, such as total number of reviews left by a reviewer, date of a review relative to when a product first became available, etc., the usefulness of applying topic modeling on spam review detection has not been fully investigated. There are two basic topic models. One is called probabilistic Latent Semantic Analysis (pLSA) [31] and the other is called Latent Dirichlet Allocation (LDA) [27]. They are both unsupervised methods. pLSA learns latent topics by performing a matrix decomposition on the term-document matrix. But LDA is a generative probabilistic model that assumes a Dirichlet prior over the latent topics. In practice, pLSA is much faster to train than LDA but has lower accuracy. That is why we choose LDA over pLSA. LDA assumes that each

document consists of a mixture of topics and each topic is a probability distribution over words. It is a document generative model that specifies a probabilistic procedure by which documents are generated.

In [26], the authors proposed a generative LDA-based topic modeling approach for fake review detection. Their approach is a variation of LDA that aims to detect subtle differences between the topic-word distributions of deceptive reviews versus truthful reviews. They used probabilistic prediction to figure out how likely a review should be treated as deceptive or truthful. However, in our research, we use LDA-based semisupervised learning to build our model for detecting deceptive and truthful reviews. We use LDA to extract relevant data from the dataset to be used as features for SVM.

The researchers in [5, 6, 24] applied n-gram based text categorization techniques for extracting all the words from the corpus to use them as features. Then they used these features in different machine learning techniques, such as SVM [6], logistic regression [24], and naive Bayes [5], to identify deceptive reviews. In our approach, we do not use all words as feature. We only use the topic words generated by LDA. We chose to use SVM because it was found to comparatively perform better than other data mining algorithms in text classification [32].

In our proposed approach, we follow a two-step process where we first extract the features using LDA topic modeling, and then use these extracted features in SVM for spam review detection.

We summarize our approach in Figure 4.1. Input to LDA is a dataset consisting of all the reviews. The LDA procedure requires the user to set two parameters specifying the number of components (n_component) and the number of top words (n_top_words). Number of components represents the total number of topics to be generated by LDA and number of top words represents the total number of top words that will be generated for each topic. Output from LDA is topic words (a.k.a. top words).

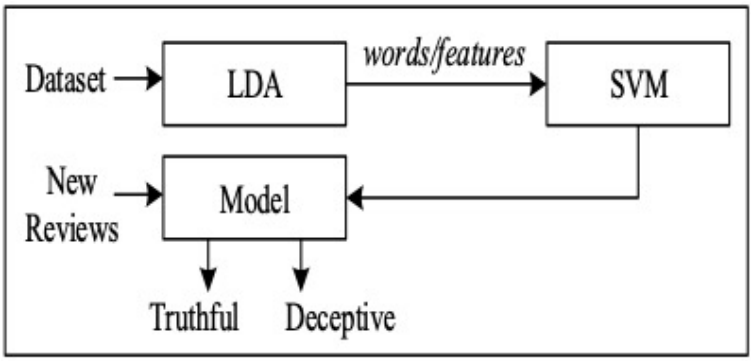

Figure 4.1: System Overview

4.1 Proposed Approach & System Implementation

In the second step, we use the topic words as features for SVM. To be more specific, a linear SVM with stochastic gradient descent is used to build the model. The new reviews are then classified by the model as truthful or deceptive.

To conduct our experiment, we used Ott et al.'s publicly available dataset of opinion spam [6]. We have described the data collection procedure in the Data Collection section.

The outline of our approach for detecting spam reviews using LDA and SVM is shown in Algorithm 2. It is similar to Algorithm 1. The most important difference is the use of LDA for topic modeling. In line 2, we utilize LDA to get the topic modeling words from all the documents. Then we use these topic words as features in SVM, instead of all the words from the training dataset.

Algorithm 2 Algorithm for detecting spam reviews using LDA and SVM

```
1: documentsList= input dataset
2: topicWordList= get topic words from documentList using LDA
3: for i = 1 to 5 do
4:     training_data, testing_data = Split documentsList
5:     tokenize = CountVectorizer(topicWordList).fit_transform(training_data)
6:     tfidf = TfidfTransformer().fit_transform(tokenize)
7:     classifier = SGDClassifier().fit(tfidf)
8:     prediction = classifier.predict(testing_data)
9: end for
```

For the LDA procedure, we set the number of components to 2, 3, or 4 because a review can be "truthful or deceptive", "truthful, deceptive, or neutral", or "truthful-positive, truthful-negative, deceptive-positive, or deceptive-negative". We started with 50 top words for each topic. Then, for each new experiment we increased the number of top words by 50 and continued this process for up to 1000 top words per topic. A sample output from LDA with 2 components and 20 top words is as follows:

Topic #1: hotel chicago room stay staff great service stayed time place recommend make like business visit friendly just city definitely enjoy

Topic #2: room hotel great stay night location bed chicago staff nice good walk clean stayed bathroom comfortable michigan view service restaurant

Below are samples of two reviews with the words that appear in either Topic #1 or Topic #2 shown in bold. The first review is a sample of a positive truthful review while the second is a sample of a positive deceptive review.

1. I was completely blown away by this **hotel**. It was magnificent. I got a **great** deal and I am so happy that I **stayed** here. Before arriving I was nervous as I had read a few bad reviews about the impact the renovation was having on peoples **stay**, for example very noisy. However, whilst the renovation was still going on and the gym was not open nor the **restaurant**, it made no difference to me. My **room** was huge, **bathroom** was spacious with excellent water pressure, **bed** was perfect and the **view** was amazing. Hotel is so close to the **great** shops of Magnificent Mile, plus a comfortable walking distance to Hancock tower and Millennium Park.

2. After recent week **stay** at the Affinia Hotels, I can definitely say i will be coming back. They offer so many in **room** amenities and services, Just a very comfortable and relaxed place to be. My most enjoyable experience at the Affinia **Hotel** was the amazing customization they offered, I would recommend Affinia hotels to anyone looking for a nice place to **stay**.

4.2 Performance

In this section, we report the deception review detection performance with our proposed approach. In Figure 4.2, we show the effect of the number of components (i.e. topics) on the accuracy. We used the whole dataset for this experiment. We observed similar accuracy for deception detection with number of topics set to 2, 3, or 4 as can be seen in Figure 4.2. However, more topics means more total number of top words (i.e. features for SVM). This can be seen in Figure 4.2 where the accuracy increases as the number of top words in a topic increases. For example, 600 top words per topic provide an accuracy of about 87%. If we use 2 topics, we will need 600 * 2 = 1200 total top words. If we use 3 topics, we will need 600 * 3 = 1800 total top words. Likewise, for 4 topics, we will need 600 * 4 = 2400 total top words. Therefore, we decided to use n_component = 2 (i.e. 2 topics), because this means fewer features and reduced dimensionality for SVM in addition to better accuracy for the same total number of top words as compared to 3 or 4 topics.

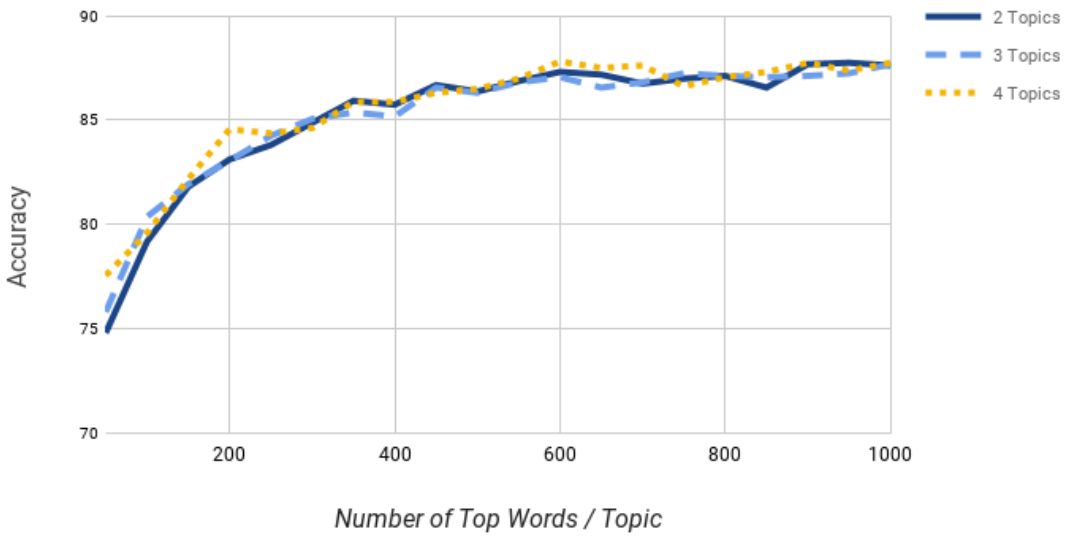

Figure 4.2: Impact of number of components on accuracy

If we look at the top words in Topic #1 and Topic #2 in the previous Section, we find some words that appear in both topics. For example “hotel” and “room” appear in both topics.

We use the term *duplicate topic words* to mean topic words that appear in both topics. We also observe the same word (e.g. stay) appearing in different forms as topic words in the same or different topics (e.g. stay and stayed both appear in Topic #2). We conducted experiments to check how these words are affecting our model's performance. We used lemmatization to convert the words in a review/document to their base forms. We preprocessed the dataset in four different ways before sending to LDA for topic modeling or before using SVM for classification as follows: 1) Without lemmatization and keeping duplicate topic words, 2) With lemmatization and keeping duplicate topic words, 3) Without lemmatization and removing duplicate topic words, and 4) With Lemmatization and removing duplicate topic words.

Figure 4.3, shows the performance of our model on the positive reviews datasets. Initially, lemmatization resulted in a slightly better performance accuracy than without lemmatization. But after about 530 top words, the accuracy is about the same or slightly better without lemmatization. The highest accuracy we receive with lemmatization is 87.37% and without lemmatization is 88.12% when the number of top words are 400 and 850, respectively. This experiment shows that lemmatization does not have much effect on improving performance. Figure 4.3 also shows that when we removed the duplicate words that appeared in both topics, our model's accuracy decreased by 5%-10%. Figure 4.4 shows similar results when we ran the experiments on the negative reviews dataset. So we proceeded with the experiment without lemmatization and without removing duplicate topic words.

Figure 4.5 and Figure 4.6 show the performance of our model on the positive reviews dataset. The accuracy varies from 84% to 88% for top words between 200 and 1000. In comparison, Ott et al. [6] achieved 89% but they considered all the words of the dataset as features with dimensionality of approximately 5476. Nonetheless, our model's accuracy is based on a maximum dimensionality of 2000.

Our model performs similarly on the negative reviews dataset as can be seen in Figure 4.7, Figure 4.8 and Table 4.1. With regards to precision, recall, and F1-score for truthful reviews

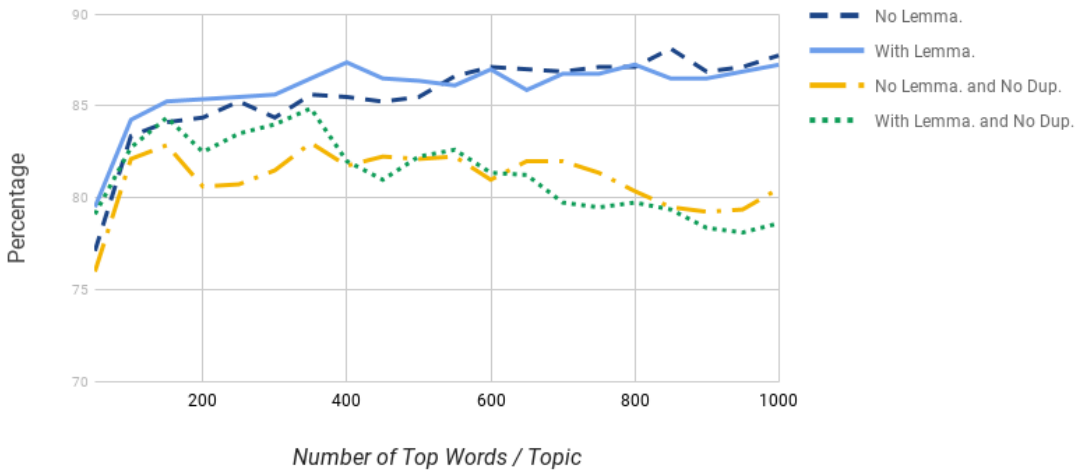

Figure 4.3: Impact of lemmatization and the presence/removal of duplicate topic words on the accuracy of the model on the **positive** reviews dataset

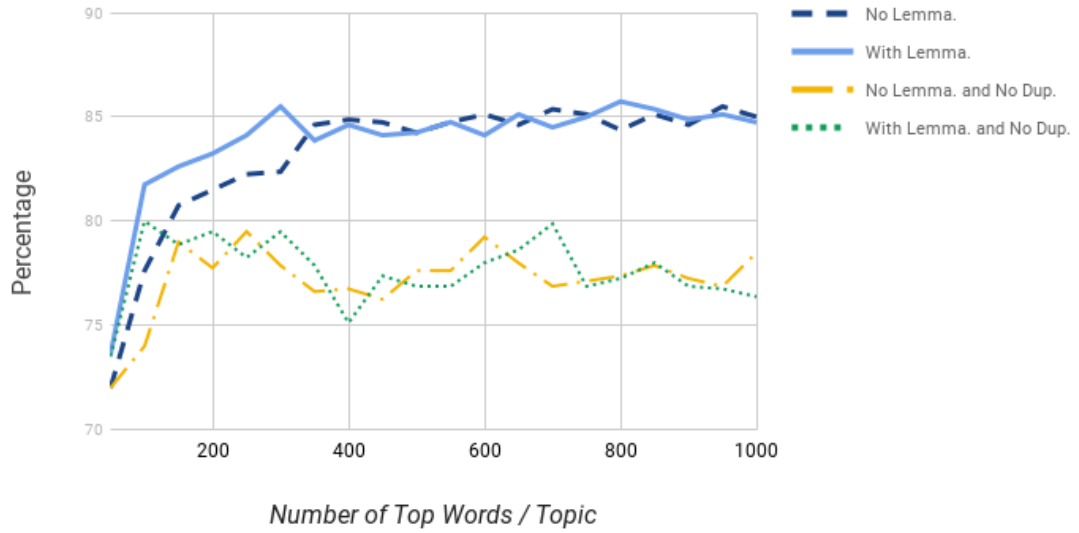

Figure 4.4: Impact of lemmatization and the presence/removal of duplicate topic words on the accuracy of the model on the **negative** reviews dataset

and deceptive reviews, Figures 4.5 and 4.6 show that increasing the number of top words contributes positively to higher precision, recall, and F1-score. This validates that our proposed model is working properly since accuracy, precision, recall, and F1-score results are consistent. Similarly, Figures 4.7 and 4.8 show the effectiveness of our model on negative reviews.

In this section, we presented an approach for spam review detection using topic model-

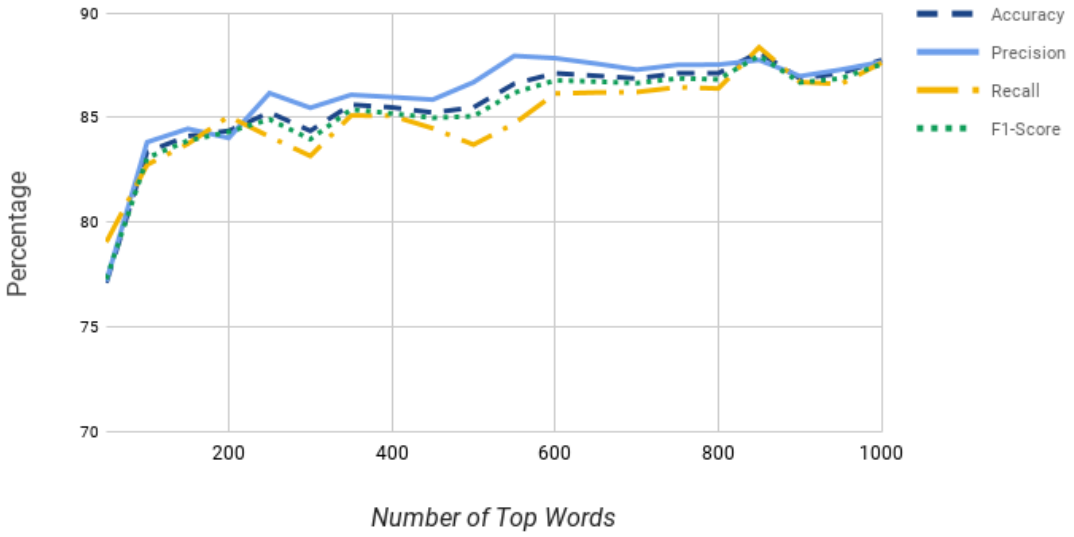

Figure 4.5: Accuracy, Precision, Recall and F1-Score of **truthful reviews** of spam review detection using topic modeling and SVM on the **positive** reviews dataset.

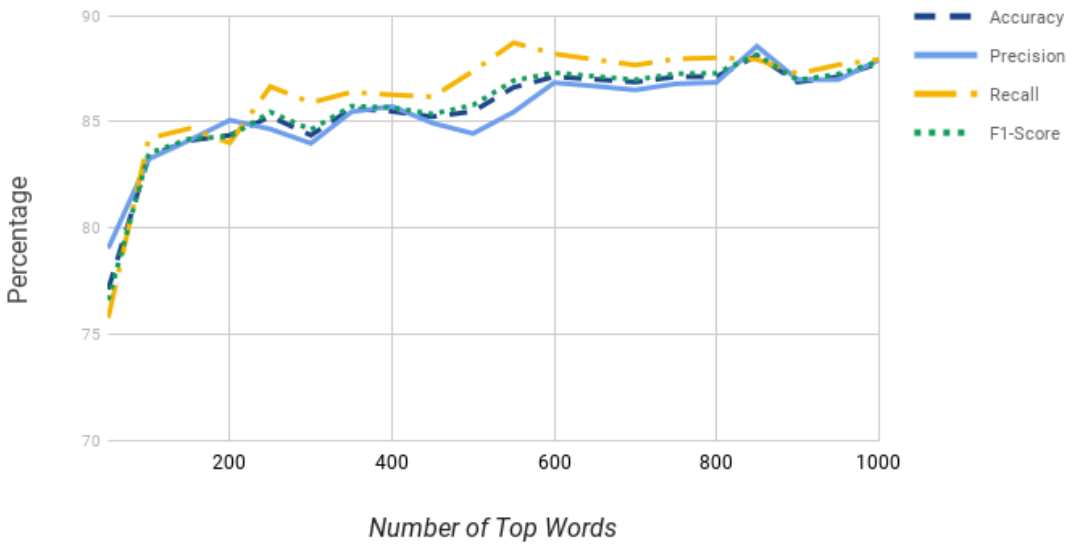

Figure 4.6: Accuracy, Precision, Recall and F1-Score of **deceptive reviews** of spam review detection using topic modeling and SVM on the **positive** reviews dataset.

ing. Our research results show that not all words in a document are needed to tag a review as truthful or deceptive. Topic modeling was used successfully to find the important words to use as features. Our approach reduces the number of features and hence dimensionality for SVM. Our model detects deceptive reviews with accuracy ranging from 84% to 88%. The accuracy we

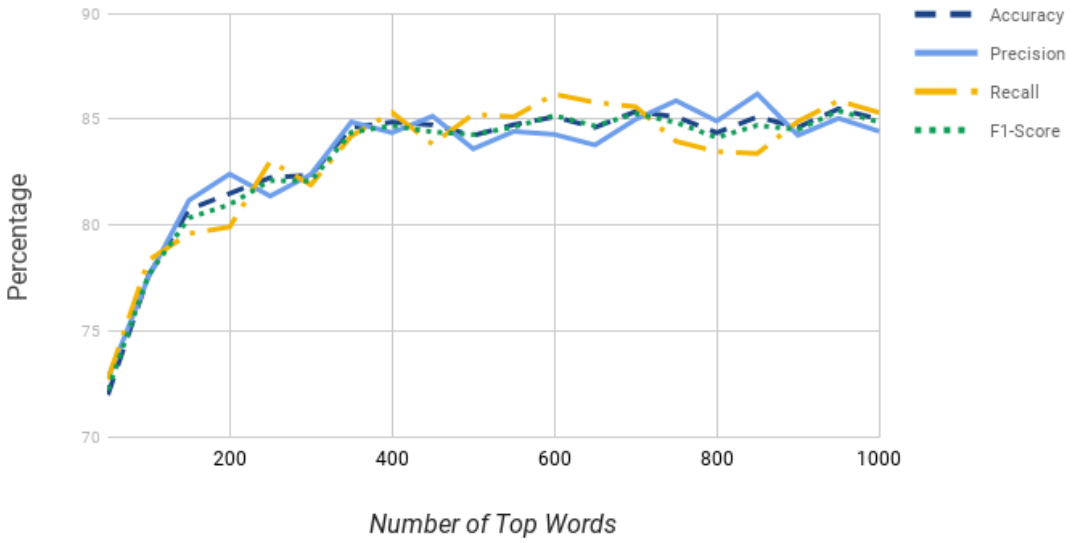

Figure 4.7: Accuracy, Precision, Recall and F1-Score of **truthful reviews** of spam review detection using Topic Modeling and SVM on the **negative** reviews data set.

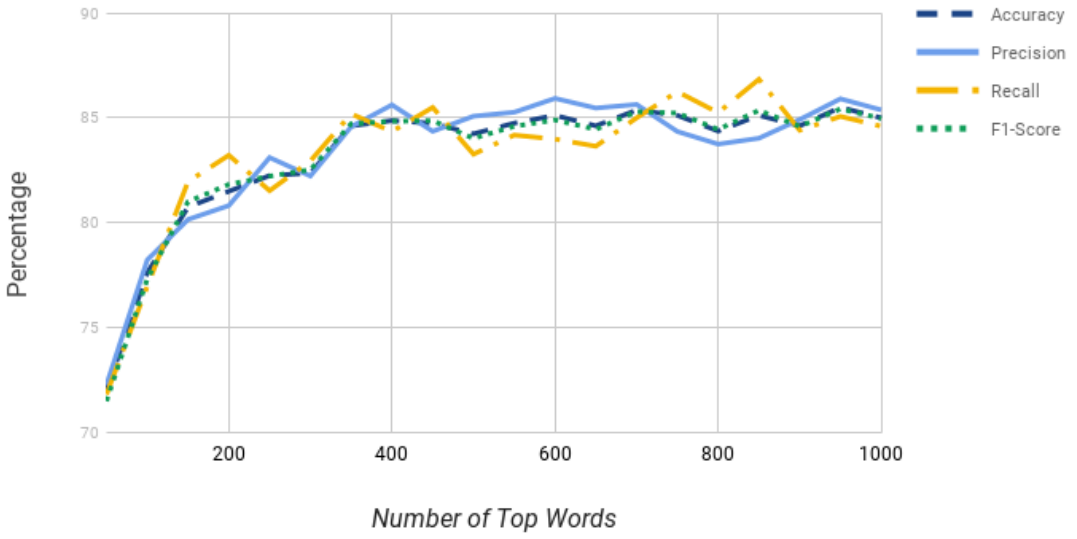

Figure 4.8: Accuracy, Precision, Recall and F1-Score of **deceptive reviews** of spam review detection using Topic Modeling and SVM on the **negative** reviews data set.

achieved is comparable with other approaches that use all the words in a dataset as features. The research results show that not all words in a document are needed to tag a review as truthful or deceptive.

Table 4.1: Classifier performance with positive and negative reviews dataset on 5-fold cross-validation experiments and reported precision, recall and F-1 score.

					TRUTHFUL			DECEPTIVE		
Dataset	Approach	Features	Number Of Top Words	Accuracy	P	R	F-1	P	R	F-1
			200	84.37	84.03	85.06	84.32	85.08	84.01	84.31
Positive			400	85.49	85.98	85.09	85.2	85.73	86.27	85.67
Reviews			600	87.12	87.84	86.15	86.78	86.84	**88.2**	87.31
Dataset	Topic		800	87.12	87.53	86.39	86.83	86.85	88.01	87.29
	Modeling	Topic	850	**88.12**	**87.75**	**88.36**	**87.96**	**88.56**	87.94	**88.16**
	plus	Words	200	81.5	82.42	79.92	80.98	80.82	83.22	81.83
Negative	SVM		400	84.87	84.37	85.33	84.41	85.62	84.35	84.84
Reviews			600	85.12	84.29	**86.19**	**85.18**	**85.93**	83.99	84.9
Dataset			800	84.37	84.92	83.48	84.14	83.75	85.25	84.45
			850	**85.12**	**86.2**	83.39	84.72	84.03	**86.84**	**85.36**

4.3 Performance with Unbalance Dataset

Most of the real world datasets are unbalanced. Which is why we check our model's performance with unbalance dataset. To create unbalance dataset, we keep all of the positive reviews and randomly select negative reviews from 10% to 90% of the negative reviews. For each time of unbalanced dataset creation, we decrease negative reviews by 10% and record our model's performance. If we look at Table 4.2, we can see the lowest accuracy is 84.53%. We get the lowest accuracy when we create unbalance dataset with 100% positive reviews and 60% negative reviews from the whole dataset. The highest accuracy is 88.54% which we get when we create unbalance dataset with 100% positive reviews and 20% negative reviews. It also gives us the highest precision of 88.54%.

Similarly, we create unbalance dataset by keeping all the negative reviews and randomly selecting positive reviews from 10% to 90% of the positive reviews. For each time of unbalanced data creation, we decrease positive reviews by 10% and record our model's performance in Table 4.3. We receive the lowest accuracy (85.14%), when we create unbalance dataset with 70% positive and 30% negative reviews. The highest accuracy is 92.30%, when we create unbalance dataset with 30% positive and 100% negative review dataset. It also provides us the highest preci-

sion which is 91.73%.

Finally, after analyzing our model's performance from Table 4.2 and Table 4.3, we can say that it performs similar to the balanced dataset. Thus, we can claim that our proposed approach works both with balance and unbalance dataset.

Table 4.2: Classifier performance with unbalanced reviews dataset with majority **positive** reviews on 5-fold cross-validation experiments and reported accuracy, precision, recall and F-1 score.

			TRUTHFUL			DECEPTIVE		
Positive review	Negative review	Accuracy	P	R	F-1	P	R	F-1
100%	90%	86.05	88.27	84.35	86.12	83.94	87.63	85.62
100%	80%	86.52	87.75	88.00	87.72	85.30	84.61	84.78
100%	70%	86.17	88.44	87.63	88.01	82.72	84.27	83.43
100%	60%	84.53	87.11	88.19	87.61	79.82	78.37	78.99
100%	50%	84.83	86.42	91.50	88.77	81.07	72.72	76.22
100%	40%	86.60	89.24	92.17	90.66	78.88	72.76	75.59
100%	30%	87.69	88.61	96.13	92.21	83.55	59.23	69.16
100%	20%	88.54	89.41	97.73	93.35	78.77	42.28	54.68

Table 4.3: Classifier performance with unbalanced reviews dataset with majority **negative** reviews on 5-fold cross-validation experiments and reported accuracy, precision, recall and F-1 score.

			TRUTHFUL			DECEPTIVE		
Positive review	Negative review	Accuracy	P	R	F-1	P	R	F-1
90%	100%	86.71	88.40	82.79	85.37	85.48	90.39	87.76
80%	100%	87.08	86.02	84.20	85.06	87.73	89.43	88.55
70%	100%	85.14	85.73	76.93	80.79	85.05	91.18	87.87
60%	100%	86.09	86.83	74.24	79.79	85.79	93.40	89.34
50%	100%	86.00	84.27	71.71	77.11	86.78	93.44	89.88
40%	100%	85.35	78.15	68.98	72.69	87.93	92.36	89.98
30%	100%	92.30	91.73	72.58	80.80	92.46	97.97	95.13
20%	100%	89.16	78.64	47.80	59.09	90.28	97.51	93.73

4.4 Performance with Only Topic Sentences

A topic sentence is a sentence that contains at least one topic word. For example, **Hotel** is a topic word. So the following sentence which is taken from a real hotel review: "*The Omni Chicago **Hotel** I am a business woman who travels a great deal out of a month, therefore, my accommodations must meet the highest standards.*" will be considered as a topic sentence. The following experiment consists of three steps. In the first step, we use topic modeling for extracting topic words from all of the reviews. In the second step, we remove all of the non-topic sentences from all the reviews.

Example: Output of topic modeling when n_component = 2 & n_top_words = 10.

***Topic* #1:** hotel room great chicago stay location nice staff stayed service

***Topic* #2:** hotel chicago stay room staff great rooms service stayed time

Before Sentence Removal: The Omni Chicago Hotel I am a business woman who travels a great deal out of a month, therefore, my accommodations must meet the highest standards. I was booked for a stay at The Omni Chicago Hotel, located in what is referred to as ' The Magnificent Mile ' in the greater Chicago area. ' Magnificent ', it was! The beautifully red-bricked sky scraper was indeed a breath- taking sight and upon entrance, I had a felling of warmth from the very hospitable welcoming staff. I was impressed with the hotels special rates offered during prime business hours and the guest rooms ranged everything from ' The Presidential Suite to The Governors Suite '. I accepted a more humble room as I would not need to spend very much time there during the day. I did stay inside most nights and the amenities were more than satisfactory. I enjoyed the very spacious exercise room and afterwards, I would take a quick dip in the pool. I toured the hotel as my niece is planning her wedding and just so happens to live close to the hotel. The ' Chagall Ballroom ', was elegant enough for such an occasion and reeked of pure luxury. I was given very adequate maps and directions to and from as my business was conducted throughout the city. That was a life saver. All in all, my experience was more than favorable and I would definitely stay there again along with recommending it to anyone.

After Sentence Removal: The Omni **Chicago Hotel** I am a business woman who travels

a **great** deal out of a month, therefore, my accommodations must meet the highest standards. I was booked for a stay at The Omni **Chicago Hotel**, located in what is referred to as ' The Magnificent Mile ' in the greater **Chicago** area. ' The beautifully red-bricked sky scraper was indeed a breath- taking sight and upon entrance, I had a felling of warmth from the very hospitable welcoming **staff**. I was impressed with the **hotels** special rates offered during prime business hours and the guest rooms ranged everything from ' The Presidential Suite to The Governors Suite '. I accepted a more humble **room** as I would not need to spend very much time there during the day. I did **stay** inside most nights and the amenities were more than satisfactory. I enjoyed the very spacious exercise **room** and afterwards, I would take a quick dip in the pool. I toured the **hotel** as my niece is planning her wedding and just so happens to live close to the hotel. All in all, my experience was more than favorable and I would definitely **stay** there again along with recommending it to anyone.

In the third step, we build a classifier/model using linear SVM with stochastic gradient descent (SGD). To build this model, we use our new dataset which we get from step 2. This generated model is used for measuring performance with testing data. We use 5-Fold cross-validation to measure performance.

Figure 4.9 and Figure 4.10 show the performance of our model on the positive reviews dataset. The accuracy varies from 84% to 88% for top words between 200 and 1000. After 600 top words, its accuracy remains above 87%. It gives a similar accuracy as Figure 4.5 and 4.6. The highest precision we get from Figure 4.9 is 87.99% and from Figure 4.10 is 87.96%. On the other hand, the lowest precision we get from Figure 4.9 is 77.65% and from Figure 4.10 we get 78.80%.

Similarly, Figure 4.11 and Figure 4.12 show the performance of our model on the negative reviews dataset. The accuracy varies from 81% to 86% for top words between 200 and 1000. After 550 top words, the accuracy remains above 85%. It gives similar accuracy as Figure 4.7 and 4.8. The highest precision we get from Figure 4.11 is 85.99% and from Figure 4.12 we get 87.19%. On the other hand, the lowest precision we get from Figure 4.11 is 72.80% and 72.72%

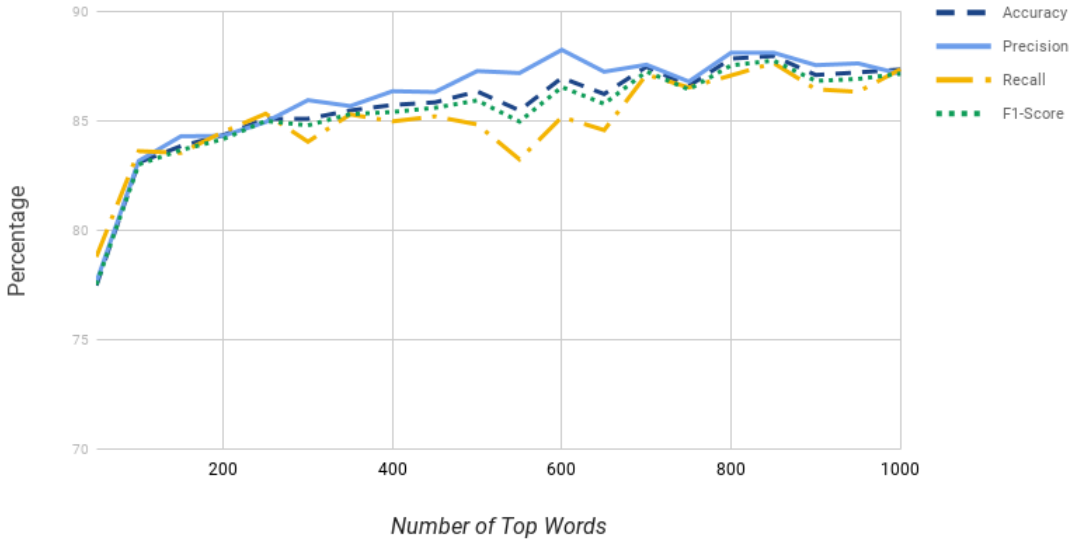

Figure 4.9: Accuracy, Precision, Recall and F1-Score of **truthful reviews** of spam review detection using Topic Modeling and SVM on the **positive** reviews with **topic sentence** data set.

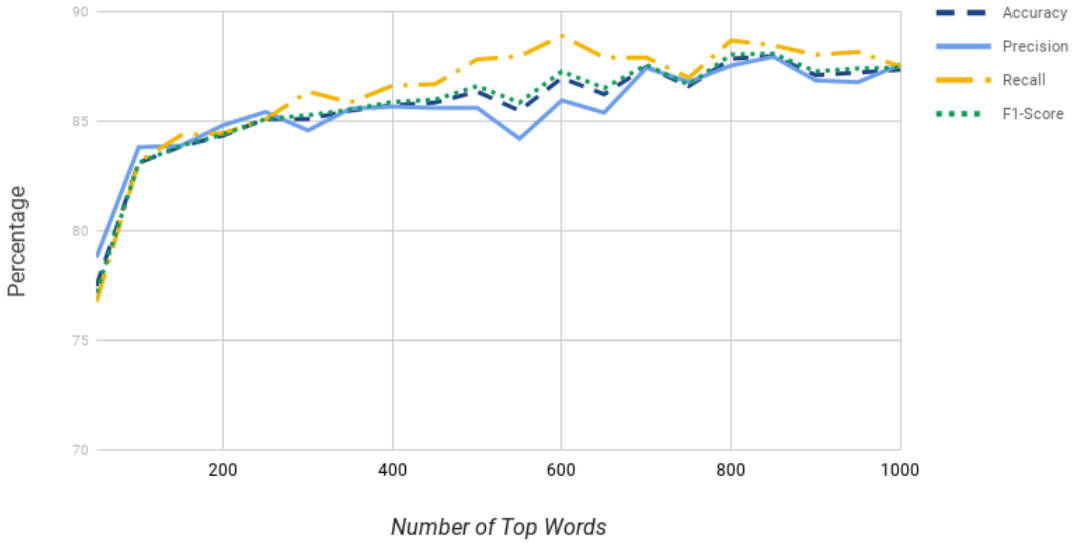

Figure 4.10: Accuracy, Precision, Recall and F1-Score of **deceptive reviews** of spam review detection using Topic Modeling and SVM on the **positive** reviews with **topic sentence** data set.

from Figure 4.12.

Comparing the performance of Figure 4.5-4.8 and Figure 4.9-4.12, we can say that non-topic word sentences do not contribute much in labeling a review as deceptive or truthful.

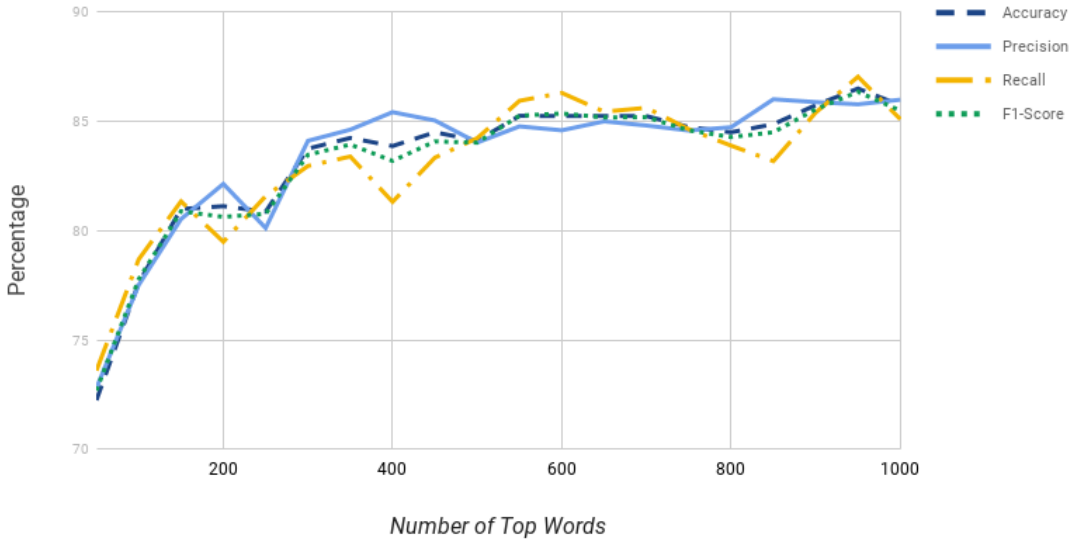

Figure 4.11: Accuracy, Precision, Recall and F1-Score of **truthful reviews** of spam review detection using Topic Modeling and SVM on the **negative** reviews with **topic sentence** data set.

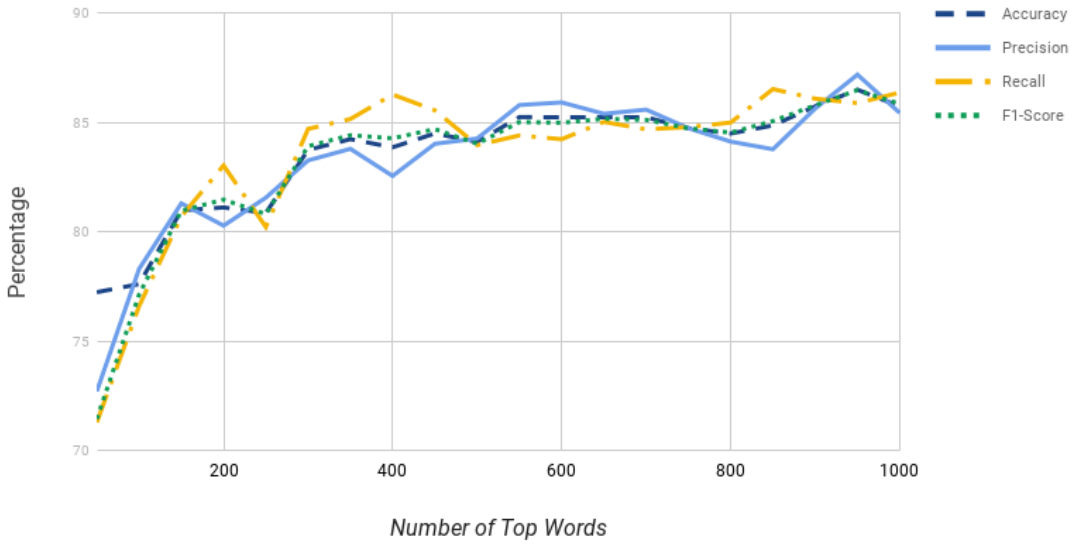

Figure 4.12: Accuracy, Precision, Recall and F1-Score of **deceptive reviews** of spam review detection using Topic Modeling and SVM on the **negative** reviews with **topic sentence** data set.

4.5 Topic Words Counting Approach

In this section, we discuss how to find the relationship between topic words and truthful/deceptive reviews. As topic modeling keeps similar types of words in the same group, we try

to explore this characteristic of topic modeling in truthful/deceptive review labeling. We want to check whether topic modeling can keep topic words from truthful reviews in one group and topic words from deceptive reviews in another group. We explore this approach because if we can get a decent result, we can label a truthful/deceptive review using only unsupervised learning techniques.

We conduct this research idea using two steps. In the first step, we extract the topic modeling words from the input dataset using LDA. We use 2 as the number of component for LDA input parameter, as we are interested in grouping the topic words into two groups. One group for truthful reviews and another group for deceptive reviews. Following is the example of topic modeling result with n_component = 2 and n_words = 50:

Topic #1: hotel like white pool small filet mignon bed good chicken doorman food dinner best did just cab restaurant wire make french lobby positive lunch service cake covers years water left emergency bit coffee bug straight lacking style child furniture street meantime served guest home monoco portions sofitel inadequate intend chair

Topic #2: room hotel stay chicago service desk did staff night rooms stayed like time just bed got check nice bathroom called told day good experience better didn great arrived location staying lobby went reservation asked hotels minutes place really finally floor took door price people clean say rude wasn said small

In the second step, we try to find out how many times each topic modeling word appears in truthful and deceptive reviews. We report the appearance of words from Topic 1 in Table 4.4 and the appearance of words from Topic 2 in Table 4.5. If we look at the appearance of topic words in Table 4.4 and Table 4.5, we can see that the frequency of the appearance of most of the topics' words is almost identical. Few words have a significant difference in appearance frequency. Words from Topic 1 which appear mostly in truthful reviews are *white*, *good*, *lobby*, *coffee*, etc. and in deceptive reviews are *like*, *food*, *make*, *service*, etc. On the other hand, words from Topic 2 which appear mostly in truthful reviews are *night*, *nice*, *told*, *day*, *location* etc. and words which appear mostly in deceptive reviews are *stay*, *service*, *desk*, *like*, *check*, *experience*, *arrived*,

etc.

After analyzing this experiment's results, we find out that for each topic most of the words appear almost the same number of times for both truthful and deceptive reviews. There are few words that appear mostly in truthful reviews and there are few word that appear mostly in deceptive reviews. Unfortunately, those small set of words are not sufficient for labeling a review as truthful/deceptive. Finally, from this experiment we can conclude that it is difficult to label a review using only the frequency that topic words appear in reviews.

Table 4.4: Appearance of topic words (from topic 1) in truthful and deceptive reviews.

Word	Appearance in Truthful Review	Appearance in Deceptive Review
hotel	330	337
like	109	151
white	14	5
pool	23	26
small	83	59
filet	1	1
mignon	1	1
bed	142	126
good	92	73
chicken	2	0
doorman	7	2
food	28	64
dinner	16	16
best	28	27
did	167	183
just	111	102
cab	21	12
restaurant	44	46
wire	14	8
make	63	87
french	2	0
lobby	83	53
positive	8	12
lunch	4	3
service	146	168
cake	5	1
covers	4	2
years	21	10
water	58	30
left	43	47
emergency	3	1
bit	39	31
coffee	42	15
bug	11	14
straight	3	4
lacking	7	9
style	8	3
child	11	13
furniture	19	20

Table 4.5: Appearance of topic words (from topic 2) in truthful and deceptive reviews.

Word	Appearance in Truthful Review	Appearance in Deceptive Review
room	340	359
hotel	330	337
stay	268	310
chicago	5	2
service	146	168
desk	118	139
did	167	183
staff	130	136
night	170	150
rooms	127	128
stayed	116	131
like	109	151
time	135	144
just	111	102
bed	142	126
got	93	107
check	122	155
nice	104	86
bathroom	99	86
called	84	71
told	87	63
day	143	103
good	92	73
experience	72	121
better	75	67
didn	75	81
great	96	52
arrived	51	100
location	107	41
staying	52	91
lobby	83	53
went	57	83
reservation	56	86
asked	67	64
hotels	71	60
minutes	54	77
place	92	100
really	53	54
finally	26	77

5 WORDS BASKET ANALYSIS

In this chapter, we describe a new approach for labeling a review as truthful/deceptive. We have named this approach *Words Basket Analysis* approach. Our approach works in a fully unsupervised fashion. We hypothesize that a word that appears more frequently in different baskets, will have a higher probability of occurring in truthful reviews. For example, if someone says, “iPhone6 has great sound quality,” and over time different reviewers keep mentioning ”Great sound quality,” then we can assume that iPhone6 has great sound quality. Therefore, we can assume those reviews stating that the iPhone6 has great sound quality are most likely to be true. If reviews about a product feature remain the same over time, then there is a high probability that they are truthful.

Research shows that online reviews usually contain more than 65% truthful reviews [11, 33]. We collect different types of Amazon product reviews, and using those reviews we check what percentage of reviews our proposed approach labels as truthful. If our approach labels most of the product reviews as containing more than 65% truthful reviews then there is a good chance that our approach is working.

5.1 Working Procedure

We present our working procedure of *Words Basket Analysis* for detecting spam reviews in Algorithm 3. First, we make a documentsList from the input dataset. In line 2, we sort the documentsList by posting time of review. Then we divide our dataset into five parts. From each part, we get top topic words using topic modeling. Then using the topic words, we create five baskets of topic words as shown in Figure 5.1. The leftmost basket contains topic words generated from the earliest set of reviews and rightmost basket contains topic words generated from the most recent set of reviews. In lines 9-14, we calculate each topic word's appearance frequency in the different baskets. In lines 17-25, we evaluate each review's probability of being labeled as truthful. In lines 18-23, we count the number of total truthful and deceptive words in a document. We

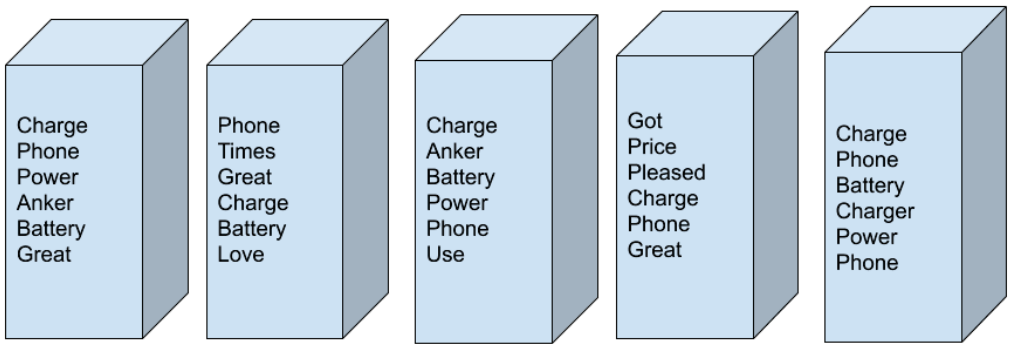

Figure 5.1: Words Basket from power bank dataset. Topic modeling is used to get the topic words in each basket.

consider a topic word a truthful word if it appears in more than a certain threshold (e.g. 3) number of baskets. Similarly, we consider a topic word as a deceptive word if it appears in less than a certain threshold (e.g. 2) number of baskets. We ignore those topic words that appear in between the two threshold values. We label them as a neutral words. In lines 26-30, we label a review as truthful or deceptive. We use the value of percentage of truthful word appearance as a parameter for labeling a review as truthful or deceptive. To label a review, we use a certain upper threshold value (e.g. 65% of truthful words appear in the review) for labeling it as truthful and a certain lower threshold value (e.g. 35% of truthful words appear in the review) for labeling it as deceptive. If a document's truthful probability is higher than the upper threshold value, then we label that document as a truthful review and if a document's truthful probability is lower than the lower threshold value then we label that document as a deceptive review.

5.2 Performance

We used a dataset consisting of five Amazon products for evaluating our proposed approach's performance. The products are Power Bank, Blending Machine, iPhone6, Book and

Algorithm 3 Algorithm for detecting spam reviews using LDA and *Words Basket Analysis* approach

```
documentsList= input dataset
sortedDocumentList= sort_by_post_time(documentsList, order = asc)
documentListArray[] = divide sortedDocumentList chronologically in five parts
wordBasketArray = []
for i = 1 to 5 do
    topicWordList = get_topic_word_using_LDA(documentListArray[i])
    wordBasketList =[]
    word_freq = {}
    for word in topicWordList do
        if word not in word_freq then
            word_freq[word] = 0
        end if
        word_freq[word] = word_freq[word] +1
    end for
    wordBasketList += topicWordList
end for
for document in documentList do
    for topic_word in document do
        if occurance(topic_word) >= upper_threshold then
            truthful_word += 1
        else if occurance(topic_word) <= lower_threshold then
            deceptive_word += 1
        end if
    end for
    probabity_of_truthful = truthful_word/(truthful_word+deceptive_word)
    if probabity_of_truthful >= upper_threshold_limit then
        label as truthful review
    else if occurance(topic_word) <= lower_threshold_limit then
        label as deceptive review
    end if
end for
```

Headphone. All of the products have a high number of reviews and their rating is higher than four. We start with an upper threshold limit greater than or equal to 60% for labeling a review as truthful and a lower threshold limit less than or equal to 40% for labeling a review as deceptive. We ignore reviews with threshold between 40% to 60% because the truthful and deceptive words appearance in this range is similar. It will be hard to distinguish truthful reviews from deceptive reviews. We are only interested in those reviews that have a significant amount of dif-

ference between truthful and deceptive word frequency. In the next two trials, we increment the upper threshold limit by five for labeling a review as truthful and decrement the lower threshold limit by five for labeling a review as deceptive.

In Table 5.1, we report our approach's performance with the Power Bank dataset. The Power Bank dataset contains 2610 reviews in total. We start with 20 topic words and increase topic words by 20 words for each iteration. We end our process with 200 topic words. For each iteration, we report total truthful, deceptive and neutral labeled reviews. As found by the *Words Basket Approach*, we also show a lower limit of truthful percentage where we consider neutral words for calculating probability of truthful percentage of a review. The formula of truthful percentage calculation is *total truthful words /(total truthful words + total deceptive words + total neutral words)*. But for showing the upper limit of the truthful percentage, we ignore neutral words for calculating probability of truthful percentage of a review. If we look at Table 5.1, we can see that with the increase of topic words, both lower and upper limit of truthful labeling percentage are increasing. The lowest lower limit of truthful labeling percentage is 73.32 and the highest lower limit of truthful labeling percentage is 95.73. The lower upper limit of truthful labeling percentage is 87.33 and the highest upper limit of truthful labeling percentage is 98.21.

Similarly, in Table 5.2, we report our approach's performance using the Blending Machine dataset. The dataset contains a total of 1000 reviews. With the increase of topic words, both the lower and upper limit of truthful labeling percentage is increasing. The lowest lower limit of truthful labeling percentage is 29.24 and the highest lower limit of truthful labeling percentage is 90.55. The lower upper limit of truthful labeling percentage is 42.23 and the highest upper limit of truthful labeling percentage is 96.26.

In Table 5.3, we show our approach's performance using the iPhone6 dataset. The dataset contains in total 2259 reviews. Here also with the increase of topic words both the lower and upper limit of truthful labeling percentage is increasing. The lowest lower limit of truthful labeling

Table 5.1: Performance of *Words Basket Approach* with a Power Bank's reviews dataset where upper threshold is 60% and lower threshold is 40% for labeling truthful and deceptive reviews.

Topic Words	Total Review	Truthful	Deceptive	Neutral	Truthful Percentage (lower limit)	Truthful Percentage (upper limit)
20	2610	1910	277	418	**73.32**	**87.33**
40	2610	2185	140	280	83.87	93.97
60	2610	2263	97	245	86.87	95.88
80	2610	2377	80	148	91.24	96.74
100	2610	2481	61	63	95.23	97.60
120	2610	2494	54	57	**95.73**	97.88
140	2610	2494	46	65	95.73	98.18
160	2610	2481	50	74	95.23	98.02
180	2610	2445	55	105	93.85	97.80
200	2610	2477	45	83	95.08	**98.21**

Table 5.2: Performance of *Words Basket Approach* with a Blending Machine's reviews dataset where upper threshold is 60% and lower threshold is 40% for labeling truthful and deceptive reviews.

Topic Words	Total Review	Truthful	Deceptive	Neutral	Truthful Percentage (lower limit)	Truthful Percentage (upper limit)
20	1000	291	398	306	**29.24**	**42.23**
40	1000	668	110	217	67.13	85.86
60	1000	728	88	179	73.16	89.21
80	1000	747	78	170	75.07	90.54
100	1000	759	64	172	76.28	92.22
120	1000	804	52	139	80.80	93.92
140	1000	817	45	133	82.11	94.77
160	1000	895	38	62	89.94	95.92
180	1000	899	35	61	90.35	96.25
200	1000	901	35	59	**90.55**	**96.26**

percentage for the iPhone6 dataset is 27.42 and the highest lower limit of truthful labeling percentage is 49.06. The lower upper limit of truthful labeling percentage is 37.96 and the highest upper limit of truthful labeling percentage is 74.24.

In Table 5.4, we present our approach's performance using the Book dataset. The dataset contains in total 1700 reviews. The lowest lower limit of truthful labeling percentage for the Book dataset is 45.84 and the highest lower limit of truthful labeling percentage is 92.86. The

Table 5.3: Performance of *Words Basket Approach* with an iPhone6's reviews dataset where upper threshold is 60% and lower threshold is 40% for labeling truthful and deceptive reviews.

Topic Words	Total Review	Truthful	Deceptive	Neutral	Truthful Percentage (lower limit)	Truthful Percentage (upper limit)
20	2259	800	969	481	35.55	45.22
40	2259	863	821	566	38.35	51.24
60	2259	617	1008	625	**27.42**	**37.96**
80	2259	780	709	761	34.66	52.38
100	2259	874	646	730	38.84	57.50
120	2259	927	603	720	41.20	60.58
140	2259	983	507	760	43.68	65.97
160	2259	1011	469	770	44.93	68.31
180	2259	1048	425	777	46.57	71.14
200	2259	1104	383	763	**49.06**	**74.24**

Table 5.4: Performance of *Words Basket Approach* with a Book's reviews dataset where upper threshold is 60% and lower threshold is 40% for labeling truthful and deceptive reviews.

Topic Words	Total Review	Truthful	Deceptive	Neutral	Truthful Percentage (lower limit)	Truthful Percentage (upper limit)
20	1700	777	423	495	**45.84**	**64.75**
40	1700	1409	108	178	83.12	92.88
60	1700	1382	113	200	81.53	92.44
80	1700	1233	134	328	72.74	90.19
100	1700	1211	127	357	71.44	90.50
120	1700	1441	79	175	85.01	94.80
140	1700	1499	64	132	88.43	95.90
160	1700	1469	60	166	86.66	96.07
180	1700	1546	54	95	91.20	96.66
200	1700	1574	45	76	**92.86**	**97.22**

lower upper limit of truthful labeling percentage is 64.75 and the highest upper limit of truthful labeling percentage is 97.22.

In Table 5.5, we outline our approach's performance using a Headphone dataset. The dataset contains in total 5000 reviews. The lowest lower limit of truthful labeling percentage for the Headphone dataset is 52.15 and the highest lower limit of truthful labeling percentage is 95.17. The lower upper limit of truthful labeling percentage is 78.67 and the highest upper limit of truthful labeling percentage is 99.47.

Table 5.5: Performance of *Words Basket Approach* with a Headphone's reviews dataset where upper threshold is 60% and lower threshold is 40% for labeling truthful and deceptive reviews.

Topic Words	Total Review	Truthful	Deceptive	Neutral	Truthful Percentage (lower limit)	Truthful Percentage (upper limit)
20	5000	2605	706	1684	**52.15**	**78.67**
40	5000	4256	135	604	85.20	96.92
60	5000	3239	232	1524	64.84	93.31
80	5000	3713	151	1131	74.33	96.09
100	5000	4363	72	560	87.34	98.37
120	5000	4363	60	572	87.34	98.64
140	5000	4537	48	410	90.83	98.95
160	5000	4615	34	346	92.39	99.26
180	5000	4649	34	312	93.07	99.27
200	5000	4754	25	216	**95.17**	**99.47**

Table 5.6: Performance of *Words Basket Approach* with a Power Bank's reviews dataset where upper threshold is 65% and lower threshold is 35% for labeling truthful and deceptive reviews.

Topic Words	Total Review	Truthful	Deceptive	Neutral	Truthful Percentage (lower limit)	Truthful Percentage (upper limit)
20	2610	1679	218	708	**64.45**	**88.50**
40	2610	1987	120	498	76.27	94.30
60	2610	2059	82	464	79.04	96.17
80	2610	2250	72	283	86.37	96.89
100	2610	2438	56	111	93.58	97.75
120	2610	2446	51	108	**93.89**	97.95
140	2610	2437	42	126	93.55	98.30
160	2610	2388	45	172	91.66	98.15
180	2610	2322	45	238	89.13	98.09
200	2610	2397	39	169	92.01	**98.39**

In Tables 5.6-5.10, we report our approach's performance with an upper threshold greater than or equal to 65% and a lower threshold smaller than or equal to 35%. In all the tables with an increase of topic words, the lower and upper limits of truthful labeling percentage usually start to increase.

Table 5.7: Performance of *Words Basket Approach* with a Blending Machine's reviews dataset where upper threshold is 65% and lower threshold is 35% for labeling truthful and deceptive reviews.

Topic Words	Total Review	Truthful	Deceptive	Neutral	Truthful Percentage (lower limit)	Truthful Percentage (upper limit)
20	1000	233	322	440	**23.41**	**41.98**
40	1000	540	95	360	54.27	85.03
60	1000	609	70	316	61.20	89.69
80	1000	624	70	301	62.71	89.91
100	1000	643	56	296	64.62	91.98
120	1000	658	41	296	66.13	94.13
140	1000	683	39	275	68.64	94.59
160	1000	884	33	118	84.82	96.23
180	1000	843	30	122	84.72	96.56
200	1000	853	30	112	**85.72**	**96.60**

Table 5.8: Performance of *Words Basket Approach* with an iPhone6's reviews dataset where upper threshold is 65% and lower threshold is 35% for labeling truthful and deceptive reviews.

Topic Words	Total Review	Truthful	Deceptive	Neutral	Truthful Percentage (lower limit)	Truthful Percentage (upper limit)
20	2259	691	865	694	30.71	44.40
40	2259	732	680	838	32.53	51.84
60	2259	485	797	968	**21.55**	**37.83**
80	2259	624	549	1077	27.73	53.19
100	2259	693	490	1067	30.8	58.57
120	2259	725	446	1079	32.22	61.91
140	2259	768	377	1105	34.13	67.07
160	2259	799	352	1099	35.51	69.41
180	2259	800	324	1126	35.55	71.17
200	2259	848	303	1099	**37.68**	**73.67**

In Tables 5.11-5.15, we show our approach's performance with an upper threshold greater than or equal to 70% and a lower threshold smaller than or equal to 30%. In all of those tables reflecting an increase in topic words, the lower and upper limit of truthful labeling percentage usually start to increase.

After analyzing all the Tables from 5.1-5.15, we can say that we get a better result for labeling a review as truthful or deceptive when we use a threshold greater than or equal to 65% for labeling a truthful review and use threshold smaller than or equal to 35% for labeling a decep-

Table 5.9: Performance of *Words Basket Approach* with a Book's reviews dataset where upper threshold is 65% and lower threshold is 35% for labeling truthful and deceptive reviews.

Topic Words	Total Review	Truthful	Deceptive	Neutral	Truthful Percentage (lower limit)	Truthful Percentage (upper limit)
20	1700	628	331	736	**37.05**	**65.48**
40	1700	1310	91	294	77.28	93.50
60	1700	1254	94	347	73.98	93.02
80	1700	1029	92	574	60.70	91.79
100	1700	987	90	618	58.23	91.64
120	1700	1300	66	329	76.69	95.16
140	1700	1382	54	259	81.53	96.23
160	1700	1311	51	333	77.34	96.25
180	1700	1452	44	199	85.66	97.05
200	1700	1488	41	166	**87.78**	**97.31**

Table 5.10: Performance of *Words Basket Approach* with a Headphone's reviews dataset where upper threshold is 65% and lower threshold is 35% for labeling truthful and deceptive reviews.

Topic Words	Total Review	Truthful	Deceptive	Neutral	Truthful Percentage (lower limit)	Truthful Percentage (upper limit)
20	5000	1898	454	2643	**37.99**	**80.69**
40	5000	3830	88	1077	76.67	97.75
60	5000	2341	126	2528	46.86	94.89
80	5000	2872	78	2045	57.49	97.35
100	5000	3856	38	1101	77.19	99.02
120	5000	3767	29	1199	75.41	99.23
140	5000	4101	22	872	82.10	99.46
160	5000	4201	21	773	84.10	99.50
180	5000	4262	19	714	85.32	99.55
200	5000	4499	17	479	**90.07**	**99.62**

tive review. Our approach consistently labels more than 65% of the reviews as truthful for each dataset. The high performance of our approach is fathomable as all of the products we use for our research have more than four-star customer review rating. Therefore, we can confidently say that our approach of labeling a review as truthful/deceptive, in fully unsupervised fashion, is working.

Table 5.11: Performance of *Words Basket Approach* with a Power Bank's reviews dataset where upper threshold is 70% and lower threshold is 30% for labeling truthful and deceptive reviews.

Topic Words	Total Review	Truthful	Deceptive	Neutral	Truthful Percentage (lower limit)	Truthful Percentage (upper limit)
20	2610	1390	167	1048	**53.35**	**89.27**
40	2610	1733	95	777	66.52	94.80
60	2610	1786	73	746	68.56	96.07
80	2610	2033	63	509	78.04	96.99
100	2610	2333	53	219	89.55	97.77
120	2610	2321	47	237	89.09	98.01
140	2610	2338	38	229	**89.75**	98.40
160	2610	2264	38	303	86.90	98.34
180	2610	2140	39	426	82.14	98.21
200	2610	2241	34	330	86.02	**98.50**

Table 5.12: Performance of *Words Basket Approach* with a Blending Machine's reviews dataset where upper threshold is 70% and lower threshold is 30% for labeling truthful and deceptive reviews.

Topic Words	Total Review	Truthful	Deceptive	Neutral	Truthful Percentage (lower limit)	Truthful Percentage (upper limit)
20	1000	177	245	573	**17.78**	**41.94**
40	1000	431	72	492	43.31	85.68
60	1000	492	55	448	49.44	89.94
80	1000	503	58	434	50.55	89.66
100	1000	506	45	444	50.85	91.83
120	1000	488	35	472	49.04	93.30
140	1000	485	34	476	48.74	93.44
160	1000	763	29	203	**76.68**	96.33
180	1000	740	26	229	74.37	96.60
200	1000	763	26	206	76.68	**96.70**

5.3 Words Basket Analysis using Useful Votes

Each Amazon product review has a *useful vote* option. If any customer finds a review of Amazon product is useful, they can give a *useful vote* to the review. In this section, we use *useful votes* of Amazon products for measuring our *Words Basket Approach's* performance. We hypothesize that a review with *useful votes* has a higher probability of being considered as truthful than a review that does not have any *useful votes*. Usually, we give other people's review *useful votes* when we find them truthful. In this section, we explore our above assumption and report

Table 5.13: Performance of *Words Basket Approach* with an iPhone6's reviews dataset where upper threshold is 70% and lower threshold is 30% for labeling truthful and deceptive reviews.

Topic Words	Total Review	Truthful	Deceptive	Neutral	Truthful Percentage (lower limit)	Truthful Percentage (upper limit)
20	2259	535	701	1014	**23.77**	**43.28**
40	2259	571	537	1142	25.37	51.53
60	2259	387	612	1251	17.20	38.73
80	2259	476	404	1370	21.15	54.09
100	2259	542	369	1339	24.08	59.49
120	2259	568	339	1343	25.24	62.62
140	2259	596	295	1359	26.48	66.89
160	2259	612	283	1355	27.20	68.37
180	2259	616	262	1372	27.37	70.15
200	2259	653	250	1347	**29.02**	**72.31**

Table 5.14: Performance of *Words Basket Approach* with a Book's reviews dataset where upper threshold is 70% and lower threshold is 30% for labeling truthful and deceptive reviews.

Topic Words	Total Review	Truthful	Deceptive	Neutral	Truthful Percentage (lower limit)	Truthful Percentage (upper limit)
20	1700	477	234	984	**28.14**	**67.08**
40	1700	1146	72	477	67.61	94.08
60	1700	1066	71	558	62.89	93.75
80	1700	791	73	831	46.66	91.55
100	1700	776	70	849	45.78	91.72
120	1700	1122	54	519	66.19	95.40
140	1700	1206	46	443	71.15	96.32
160	1700	1111	43	541	65.54	96.27
180	1700	1276	39	380	75.28	97.03
200	1700	1358	40	297	**80.11**	**97.13**

our *Words Basket Approach's* performance.

From the previous section, we know that our *Words Basket Approach* gives the best performance when we use an upper threshold limit of 65% for labeling truthful reviews and a lower threshold limit of 35% for labeling deceptive reviews. Thus, for our experiment, we set the upper threshold limit to 65% and lower threshold limit to 35%. In this approach, we follow the same procedure as in Algorithm 3. We only add one new statement before Line 27 and another new statement before Line 29. If a review has a truthful probability greater than or equal to the upper

Table 5.15: Performance of *Words Basket Approach* with a Headphone's reviews dataset where upper threshold is 70% and lower threshold is 30% for labeling truthful and deceptive reviews.

Topic Words	Total Review	Truthful	Deceptive	Neutral	Truthful Percentage (lower limit)	Truthful Percentage (upper limit)
20	5000	1386	307	3302	**27.74**	**81.86**
40	5000	3264	56	1675	65.34	98.31
60	5000	1510	73	3412	30.23	95.38
80	5000	1969	53	2973	39.41	97.37
100	5000	3161	21	1813	63.28	99.34
120	5000	2885	20	2090	57.75	99.31
140	5000	3364	15	1616	67.34	99.55
160	5000	3451	15	1529	69.08	99.56
180	5000	3575	15	1405	71.57	99.58
200	5000	3955	12	1028	**79.17**	**99.69**

threshold value and also has at least one *useful vote*, then we label it as correctly truthful labeled. Next, we calculate the total percentage of correctly truthful labeled reviews. On the other hand, if a review has a truthful probability less than or equal to the lower threshold value and also has zero *useful votes*, we label it as correctly deceptive labeled. Then, we calculate the total percentage of correctly deceptive labeled reviews. In Table 5.16-5.20, we outline our *Words Basket Approach's* performance using *useful votes* as the baseline.

In Table 5.16, we outline our approach's performance using the Power Bank dataset. The lowest correctly truthful labeling percentage is 17.68 and the highest correctly truthful labeling percentage 19.44. The lowest correctly deceptive labeling percentage for the Power Bank dataset is 82.22 and the highest correctly deceptive labeling percentage is 90.27.

In Table 5.17, we report our approach's performance using the Blending Machine dataset. The lowest correctly truthful labeling percentage is 36.48 and the highest correctly truthful labeling percentage 56.27. The lowest correctly deceptive labeling percentage for the Blending Machine dataset is 51.24 and the highest correctly deceptive labeling percentage is 69.69.

In Table 5.18, we show our approach's performance using the iPhone6 dataset. The lowest correctly truthful labeling percentage is 33.40 and the highest correctly truthful labeling percentage 37.04. The lowest correctly deceptive labeling percentage for the Blending Machine dataset

is 63.81 and the highest correctly deceptive labeling percentage is 70.29.

In Table 5.19, we report our approach's performance using the Book dataset. The lowest correctly truthful labeling percentage is 21.81 and the highest correctly truthful labeling percentage 28.94. The lowest correctly deceptive labeling percentage for the iPhone6 dataset is 78.24 and the highest correctly deceptive labeling percentage is 90.90.

In Table 5.20, we outline our approach's performance using the Headphone dataset. The lowest correctly truthful labeling percentage is 19.86 and the highest correctly truthful labeling percentage 23.13. The lowest correctly deceptive labeling percentage for the Headphone dataset is 57.89 and the highest correctly deceptive labeling percentage is 75.99.

After analyzing all the tables from 5.16 - 5.20, we can say that our *Words Basket Approach* does not provide accurate result in labeling truthful reviews; however, it performs decently in terms of labeling deceptive reviews. From the performance of deceptive labeling, we see that most of the deceptive reviews do not have any *useful votes*. This makes sense, because people do not grant *useful votes* to a post that they think might be a fake review. On the other hand, based off of the performance of truthful labeling, we can say that truthful reviews may or may not have *useful votes*.

As our *Words Basket Approach* is performing poorly in labeling truthful reviews, we believe a large number of truthful reviews does not have *useful votes*. So instead of checking what percentage of truthful labeled reviews have at least one *useful vote* for labeling a review as truthful, we check what percentage of reviews with at least one *useful vote* is labeled as truthful by our *Words Basket Approach*. Here, we assume that review with *useful votes* is most likely to be a truthful review.

In Table 5.21, we outline our approach's performance using the Power Bank dataset. The dataset contains 501 reviews with at least one *useful vote*. The highest accuracy we get in la-

Table 5.16: Performance of *Words Basket Approach* using *useful votes* with a Power Bank's reviews dataset where upper threshold is 65% and lower threshold is 35% for labeling truthful and deceptive reviews.

Topic Words	Truthful Labeled	Deceptive Labeled	Correctly Truthful Labeled	Correctly Deceptive Labeled	Correctly Truthful Labeled (Percentage)	Correctly Deceptive Labeled (Percentage)
20	1679	218	297	191	**17.68**	87.61
40	1987	120	375	105	18.87	87.50
60	2059	82	383	73	18.60	89.02
80	2250	72	428	65	19.02	**90.27**
100	2538	56	459	49	18.82	87.5
120	2446	51	51	44	19.01	86.27
140	2437	42	464	36	19.03	85.71
160	2388	45	458	38	19.17	84.44
180	2322	45	448	448	19.29	**82.22**
200	2397	39	466	33	**19.44**	84.61

Table 5.17: Performance of *Words Basket Approach* using *useful votes* with a Blending Machine's reviews dataset where upper threshold is 65% and lower threshold is 35% for labeling truthful and deceptive reviews.

Topic Words	Truthful Labeled	Deceptive Labeled	Correctly Truthful Labeled	Correctly Deceptive Labeled	Correctly Truthful Labeled (Percentage)	Correctly Deceptive Labeled (Percentage)
20	233	322	85	165	**36.48**	**51.24**
40	540	95	253	53	46.85	55.78
60	609	70	309	45	50.73	64.28
80	624	70	313	41	50.16	58.57
100	643	56	326	34	50.69	60.71
120	658	41	346	24	52.58	58.53
140	683	39	373	25	54.61	64.10
160	844	33	475	23	**56.27**	**69.69**
180	843	30	471	20	55.87	66.66
200	853	30	473	19	55.45	63.33

beling truthful review is 93.01%. We get similar results with the other datasets apart from the iPhone6 dataset. The highest accuracy we receive with the Blending Machine dataset is 85.74%,

Table 5.18: Performance of *Words Basket Approach* using *useful votes* with an iphone6's reviews dataset where upper threshold is 65% and lower threshold is 35% for labeling truthful and deceptive reviews.

Topic Words	Truthful Labeled	Deceptive Labeled	Correctly Truthful Labeled	Correctly Deceptive Labeled	Correctly Truthful Labeled (Percentage)	Correctly Deceptive Labeled (Percentage)
20	691	865	256	552	**37.04**	**63.81**
40	732	680	264	444	36.06	65.29
60	485	797	162	519	**33.40**	65.11
80	624	549	216	367	34.61	66.84
100	693	490	241	330	34.77	67.34
120	725	446	253	305	34.89	68.38
140	768	377	277	261	36.06	69.23
160	799	352	287	244	35.91	69.31
180	800	324	281	224	35.12	69.13
200	848	303	299	213	35.25	**70.29**

Table 5.19: Performance of Words *Basket Approach* using *useful votes* with a Book's reviews dataset where upper threshold is 65% and lower threshold is 35% for labeling truthful and deceptive reviews.

Topic Words	Truthful Labeled	Deceptive Labeled	Correctly Truthful Labeled	Correctly Deceptive Labeled	Correctly Truthful Labeled (Percentage)	Correctly Deceptive Labeled (Percentage)
20	628	331	137	259	**21.81**	**78.24**
40	1310	91	378	72	28.85	79.12
60	1254	94	363	77	**28.94**	81.91
80	1029	92	284	74	27.59	80.43
100	987	90	259	76	26.24	84.44
120	1300	66	367	57	28.23	86.36
140	1382	54	389	47	28.14	87.03
160	1311	51	372	46	28.37	90.19
180	1452	44	416	40	28.65	**90.90**
200	1488	41	427	37	28.69	90.24

with the iPhone6 dataset is 49.29%, with the Book dataset is 89.14%, and with the HeadPhone dataset is 90.55%. After analyzing the above results, we can say that *Words Basket Approach* is

Table 5.20: Performance of *Words Basket Approach* using *useful votes* with a Headphone's reviews dataset where upper threshold is 65% and lower threshold is 35% for labeling truthful and deceptive reviews.

Topic Words	Truthful Labeled	Deceptive Labeled	Correctly Truthful Labeled	Correctly Deceptive Labeled	Correctly Truthful Labeled (Percentage)	Correctly Deceptive Labeled (Percentage)
20	1898	454	394	345	20.75	**75.99**
40	3830	88	847	59	22.11	67.04
60	2341	126	465	90	**19.86**	71.42
80	2872	78	580	52	20.19	66.66
100	3856	38	865	24	22.43	63.15
120	3767	29	841	18	22.32	62.06
140	4101	22	934	15	22.77	68.18
160	4201	21	954	14	22.70	66.66
180	4262	19	986	11	**23.13**	**57.89**
200	4499	17	1036	10	23.02	58.82

also showing promising results in labeling truthful reviews.

Table 5.21: Performance of *Words Basket Approach* using *useful votes* with a Power Bank's reviews dataset where upper threshold is 65% and lower threshold is 35% for labeling only truthful reviews.

Topic Words	Total Review with at least one Useful Votes	Correctly Truthful Labeled	Correctly Truthful Labeled (Percentage)
20	501	297	59.28
40	501	375	74.85
60	501	383	76.44
80	501	428	85.42
100	501	459	91.61
120	501	465	92.81
140	501	464	92.61
160	501	458	91.41
180	501	448	89.42
200	501	466	**93.01**

6 CONCLUSION

In our research, we took different approaches for spam review detection. We started with supervised method, then tried with semi-supervised method and finally, we used a fully unsupervised method for spam review detection.

First, we applied different supervised data mining algorithms such as Support Vector Machine, Naive Bayes and Multi-layer Perceptron. We found out overall Support Vector Machine gives better results than Naive Bayes and Multi-layer Perceptron in detecting fake reviews. We simulated Ott et al.'s [6] work and obtained similar results. It gave us almost 90% accuracy in spam review detection with SVM. In addition, we used Naive Bayes algorithm which offered almost 87% accuracy and Multi-layer Perceptron offered almost 88% accuracy in spam review detection. We also looked for a relationship between Parts of Speech (POS) and truthful/deceptive reviews. But unfortunately, we did not find any distinguishable relationship between POS and truthful/deceptive reviews.

Next we worked with our newly proposed semi-supervised algorithm. We called our approach topic modeling based spam review detection. We applied a combination of topic modeling and SVM for detecting spam review. We utilized topic modeling words as features for SVM. Our proposed approach offered similar performance like Ott et al. [6] despite using only the topic words as features for SVM whereas Ott et al. [6] used all the words in a review. Our approach reduced the dimensionality for SVM and gave almost 89% accuracy in spam review detection. In addition, our model also performed well with unbalanced dataset and gave similar accuracy like balanced dataset. We also analyzed our model's performance with topic sentences. It obtained almost 89% accuracy. It implies that non-topic sentences do not contribute as a deciding factor for labeling a review as truthful or deceptive. Using topic words counting approach we tried to check whether or not topic modeling keeps truthful and deceptive topic words in different group or in the same group. We found out topic modeling does not keep truthful words in one group and de-

ceptive words in another group.

Lastly, we proposed a fully unsupervised algorithm named as *Words Basket Analysis* for detecting spam review detection. We used five different product's datasets for the performance analysis. Our approach found out all the five products contain more than 65% truthful reviews. Our model's performance is based on the claims in [11, 33] about one third of the online reviews are fake. Later, we utilized *useful votes* of Amazon review as an indicator for labeling truthful and deceptive reviews. We found out most of the deceptive reviews do not have any *useful votes*, but a truthful review may or may not have *useful votes*. We also found that reviews with *useful votes* are most likely to be labeled truthful by our *Words Basket Approach*.

For future work, we can investigate other methods for validating the *Words Basket Analysis* approach. One possibility is to create a small dataset from Amazon products reviews and manually label these reviews. Then we can validate the performance of the *Words Basket Analysis* approach performance using this manually labeled dataset. We can also use a combination of *Words Basket Analysis* and behavioral approach for labeling truthful and deceptive reviews.

REFERENCES

[1] B. Liu, "Sentiment analysis and opinion mining," *Synthesis lectures on human language technologies*, vol. 5, no. 1, pp. 1–167, 2012.

[2] M. Ott, C. Cardie, and J. Hancock, "Estimating the prevalence of deception in online review communities," in *Proceedings of the 21st International Conference on World Wide Web*, pp. 201–210, ACM, 2012.

[3] B. Liu, *Sentiment analysis: Mining opinions, sentiments, and emotions*. Cambridge University Press, 2015.

[4] N. Jindal and B. Liu, "Review spam detection," in *Proceedings of the 16th international conference on World Wide Web*, pp. 1189–1190, ACM, 2007.

[5] F. H. Li, M. Huang, Y. Yang, and X. Zhu, "Learning to identify review spam," in *Twenty-Second International Joint Conference on Artificial Intelligence*, 2011.

[6] M. Ott, Y. Choi, C. Cardie, and J. 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*, pp. 309–319, Association for Computational Linguistics, 2011.

[7] S. Feng, R. Banerjee, and Y. Choi, "Syntactic stylometry for deception detection," in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, pp. 171–175, Association for Computational Linguistics, 2012.

[8] G. Wang, S. Xie, B. Liu, and P. S. Yu, "Review graph based online store review spammer detection," in *Proceedings of the 2011 IEEE 11th International Conference on Data Mining*, pp. 1242–1247, IEEE Computer Society, 2011.

[9] M. Ott, C. Cardie, and J. T. Hancock, "Negative deceptive opinion spam," in *Proceedings of the 2013 conference of the north american chapter of the association for computational linguistics: human language technologies*, pp. 497–501, 2013.

[10] V. Sandulescu and M. Ester, "Detecting singleton review spammers using semantic similarity," in *Proceedings of the 24th international conference on World Wide Web*, pp. 971–976, ACM, 2015.

[11] M. Luca and G. Zervas, "Fake it till you make it: Reputation, competition, and yelp review fraud," *Management Science*, vol. 62, no. 12, pp. 3412–3427, 2016.

[12] E. D. Wahyuni and A. Djunaidy, "Fake review detection from a product review using modified method of iterative computation framework," in *MATEC Web of Conferences*, vol. 58, EDP Sciences, 2016.

[13] N. Jindal, B. Liu, and E.-P. Lim, "Finding unusual review patterns using unexpected rules," in *Proceedings of the 19th ACM international conference on Information and knowledge management*, pp. 1549–1552, ACM, 2010.

[14] E.-P. Lim, V.-A. Nguyen, N. Jindal, B. Liu, and H. W. Lauw, "Detecting product review spammers using rating behaviors," in *Proceedings of the 19th ACM international conference on Information and knowledge management*, pp. 939–948, ACM, 2010.

[15] A. Mukherjee, V. Venkataraman, B. Liu, and N. Glance, "What yelp fake review filter might be doing?," in *Seventh international AAAI conference on weblogs and social media*, 2013.

[16] H. Li, Z. Chen, A. Mukherjee, B. Liu, and J. Shao, "Analyzing and detecting opinion spam on a large-scale dataset via temporal and spatial patterns," in *ninth international AAAI conference on web and social Media*, 2015.

[17] S. Xie, G. Wang, S. Lin, and P. S. Yu, "Review spam detection via temporal pattern discovery," in *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 823–831, ACM, 2012.

[18] H. Li, G. Fei, S. Wang, B. Liu, W. Shao, A. Mukherjee, and J. Shao, "Bimodal distribution and co-bursting in review spam detection," in *Proceedings of the 26th International Conference on World Wide Web*, pp. 1063–1072, International World Wide Web Conferences Steering Committee, 2017.

[19] S. KC and A. Mukherjee, "On the temporal dynamics of opinion spamming: Case studies on yelp," in *Proceedings of the 25th International Conference on World Wide Web*, pp. 369–379, International World Wide Web Conferences Steering Committee, 2016.

[20] R. Shebuti and L. Akoglu, "Collective opinion spam detection: bridging review networksand metadata," in *ACM KDD*, 2015.

[21] B. Hooi, N. Shah, A. Beutel, S. Günnemann, L. Akoglu, M. Kumar, D. Makhija, and C. Faloutsos, "Birdnest: Bayesian inference for ratings-fraud detection," in *Proceedings of the 2016 SIAM International Conference on Data Mining*, pp. 495–503, SIAM, 2016.

[22] H. Li, Z. Chen, B. Liu, X. Wei, and J. Shao, "Spotting fake reviews via collective positive-unlabeled learning," in *2014 IEEE International Conference on Data Mining*, pp. 899–904, IEEE, 2014.

[23] L. Akoglu, R. Chandy, and C. Faloutsos, "Opinion fraud detection in online reviews by network effects," in *Seventh international AAAI conference on weblogs and social media*, 2013.

[24] N. Jindal and B. Liu, "Opinion spam and analysis," in *Proceedings of the 2008 international conference on web search and data mining*, pp. 219–230, ACM, 2008.

[25] S. K. Chauhan, A. Goel, P. Goel, A. Chauhan, and M. K. Gurve, "Research on product review analysis and spam review detection," in *2017 4th International Conference on Signal Processing and Integrated Networks (SPIN)*, pp. 390–393, IEEE, 2017.

[26] J. Li, C. Cardie, and S. Li, "Topicspam: a topic-model based approach for spam detection," in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, vol. 2, pp. 217–221, 2013.

[27] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of machine Learning research*, vol. 3, no. Jan, pp. 993–1022, 2003.

[28] S. B. Bhushan and A. Danti, "Classification of text documents based on score level fusion approach," *Pattern Recognition Letters*, vol. 94, pp. 118–126, 2017.

[29] L. Lenc and T. Hercig, "Neural networks for sentiment analysis in czech.," in *ITAT*, pp. 48–55, 2016.

[30] M. Ott, C. Cardie, and J. Hancock, "Estimating the prevalence of deception in online review communities," in *Proceedings of the 21st international conference on World Wide Web*, pp. 201–210, ACM, 2012.

[31] T. Hoffman, "Probabilistic latent semantic analysis," in *proc. of the 15th Conference on Uncertainty in AI, 1999*, 1999.

[32] T. Joachims, "Text categorization with support vector machines: Learning with many relevant features," in *European conference on machine learning*, pp. 137–142, Springer, 1998.

[33] D. Streitfeld, "Best book reviews money can buy," *http://nyti.ms/1cvg5b1*, 2012.

APPENDICES

Appendix A. Datasets

1. TripAdvisor Dataset:

   `https://tinyurl.com/y5r2s64h`

2. Amazon Dataset:

   `https://tinyurl.com/y6qw6cw7`

Appendix B. Codes

1. Classification:

   https://tinyurl.com/y3wbrs9b

2. Web Scrapping:

   https://tinyurl.com/y3v5yrmw