Detecting Truthful and Useful Consumer Reviews for Products using Opinion Mining

Kalpana Algotar and Ajay Bansal

Arizona State University, Mesa AZ 85212 USA
{kalgotar, ajay.banssal}@asu.edu

Abstract. Individuals and organizations rely heavily on social media these days for consumer reviews in their decision-making on purchases. However, for personal gains such as profit or fame, people post fake reviews to promote or demote certain target products as well as to deceive the reader. To get genuine user experiences and opinions, there is a need to detect such spam or fake reviews. This paper presents a study that aims to detect truthful, useful reviews and ranks them. An effective supervised learning technique is proposed to detect truthful and useful reviews and rank them, using a 'deceptive' classifier, 'useful' classifier, and a 'ranking' model respectively. Deceptive and non-useful consumer reviews from online review communities such as amazon.com and Epinions.com are used. The proposed method first uses the 'deceptive' classifier to find truthful reviews followed by the 'useful' classifier to find whether a review is useful or not. Manually labeling individual reviews is very difficult and time consuming. We incorporate a dictionary that makes it easy to label reviews. We present the experimental results of our proposed approach using our dictionary with 'deceptive' classifier and 'useful' classifier.

Keywords: Text Classification, Spam Review Detection, Opinion Mining, Supervised Learning.

1 Introduction

Nowadays, consumers looking to buy a product increasingly rely on user-generated online reviews to make or reverse their purchase decisions. Positive reviews of a product greatly influence the person's decision to buy the product. However, if one sees many negative reviews, he/she will most likely choose a different product. The outcome of positive reviews gives significant profit and advertizing for the seller and their organization. This in turn creates a market for incentivizing opinion spam. This has resulted in more and more people trying to game the system by writing fake reviews to harm or promote some products or services. A fake review means that it is either a positive review written by the business owners themselves (or people they contract to write reviews) or a negative review written by a business's competitors. Those fake reviews try to deliberately mislead readers by giving fake reviews to some entities (e.g. products) in order to promote them or to damage their reputation.

Opinion spamming refers to writing fake reviews that try to deliberately mislead human readers. The focus of spam research in the context of online reviews has been primarily on detection. Cornell University has developed a model to spot fake, non-fake review for hotels [3] as well as some existing works have been done by other researchers to detect fake reviews and spam reviewers. Recent studies, however, show that opinion spam is not easily identified by human readers [9]. In particular, humans have a difficult time identifying deceptive messages from consumer reviews. We decided to work on the same issue for product by taking different approach to make the process easier. In this approach, we choose Cornell model [3] as a base to prepare our own dictionary for fake, non-fake reviews. Our, automated approach has emerged to reliably label reviews as truthful vs. deceptive as well as second approach to label useful vs. not-useful using reader's rating on consumer's review. We train SVM text classifier using a corpus of truthful and deceptive as well as useful and not-useful reviews from Amazon and Epinion. We applied our approach to the domain of camera reviews and present the results.

The rest of the document is organized as follows: Section 2 presents related work. Background material related to this project is presented in Section 3. Our proposed approach and its implementation is presented in Section 4. Section 5 presents the experiments and analysis followed by conclusions and future work in Section 6.

2 Related Work

Web spam and email spam have been investigated extensively. The objective of Web spam is to make search engines rank the target pages high in order to attract people to visit these pages. Web spam can be categorized into two main types: content spam and link spam. Link spam is spam on hyperlinks that are placed between pages, which does not exist in reviews as usually there are no links within them. Content spam tries to add irrelevant or remotely relevant words in target pages to fool search engines to rank the target pages high. Another related research is email spam [5, 8, 14], which is also quite different from review spam. Email spam usually refers to unsolicited commercial advertisements. Although this exists, advertisements in reviews are not as frequent as in emails. They are also relatively easy to detect. Deceptive opinion spam is much harder to deal with. We present below, different approaches taken opinion spam detection.

2.1 Review Spam Detection

A preliminary study was reported in [8] to study spam review and spam detection based on finding duplicates and classification. That study proposed to treat duplicate reviews as positive training examples (with label fake), and the rest of the reviews as the negative training examples (with label non-fake). For the rest of spam (fake) reviews, they detected based on 2-class classification (spam and non-spam). In addition, they found that 52% of the highly ranked non-duplicate reviews had more than 1800 words, much higher than the average length of a normal review, and were regarded as spam reviews. A more in-depth investigation was given in [6] where three types of spam review were identified, namely untruthful reviews (reviews that promote or demote products), reviews on brands but not products, and non-reviews (e.g., advertisements). By representing a review using a set of review, reviewer and product-level features, classification techniques were used to assign spam (fake) labels to

reviews. In particular, untruthful review detection is performed by using duplicate reviews as the positive training examples (fake) and the rest of the reviews as negative training examples (non-fake) and for rest of the types manual labeling was done. In [16] neural network based model was used for representation learning of reviews.

2.2 Reviewer Spam Detection

Some of the related research addresses the problem of review spammer detection, or finding users who are the source of spam reviews. Reviews usually come with ratings. Detecting unfair ratings has been studied in several works including [4, 10]. The techniques used include: (a) clustering ratings into unfairly high ratings and unfairly low ratings, and (b) using third party ratings on the producers of ratings and ratings from less reputable producers are then deemed as unfair. Once unfair ratings are found, they can be removed to restore a fair item evaluation system. These works did not address review spammer detection directly on the reviews. They usually did not conduct evaluation of their techniques on real data.

2.3 Helpful Review Detection and Prediction

Review helpfulness prediction is closely related to review spam detection described in above. A helpful review is one that is informative and useful to the readers. The purpose of predicting review helpfulness is to faciliate review sites to provide feedback to the review contributors and to help readers choose and read high quality reviews. A classification approach to solving helpfulness prediction using review content and meta-data features was developed in [7]. The meta-data features used are review's rating and the difference between the review rating and the average rating of all reviews of the product. Liu et. al proposes to derive features from reviews content that correspond to informativeness, readability, and subjectiveness aspects of the review [9]. These features are then used to train a review helpfulness classification method.

Amazon.com allows users to vote if a review is helpful or not. These helpfulness votes are manually assigned and are thus subjective and possibly abused. Danescu-Niculescu-Mizil et. al found that a strong correlation between proportion of helpful votes of reviews and the deviation of the review ratings from the average ratings of products [3]. This correlation illustrates that helpful votes are generally consistent with average ratings. The study is however conducted at the collection level and does not provide evidence to link spam and helpfulness votes. Ott and others [11] presented a framework for estimating the prevalence of deception in online review communities. In this task, they paid one US dollar ($1) to each of 400 unique Mechanical Turk workers to write a fake positive (5-star) review for one of the 20 most heavily-reviewed Chicago hotels on TripAdvisor. For consistency with labeled deceptive review data, they simply labeled as truthful all positive (5-star) reviews of the 20 previously chosen Chicago hotels.

Detecting spam and predicting helpfulness are two separate problems since not-useful reviews are not necessarily fake. A poorly written review may be not-useful but is not fake. Spam reviews usually target specific products while not-useful votes may be given to any products. Given the motive driven nature of spamming activities, review spam detection will therefore require an approach different from not-useful review detection. Our proposed technique aims to detect truthful, useful reviews and provide a ranking of the reviews.

3 Background

3.1 Supervised Learning Methods:

A computer system learns from training data that represents some "past experiences" of an application domain. In this section, we briefly describe the various classification methods used in order to categorize reviews into deceptive, truthful and useful, not-useful. Classification involves labeling of the data (observations, measurements) with pre-defined classes. We have used three supervised learning algorithms: Support Vector Machine, Naïve Bayes, and K-Nearest Neighbor.

Support Vector Machines:

Support Vector Machines [1] are supervised learning methods used for classification, as well as regression. The advantage of Support Vector Machines is that they can make use of certain kernels in order to transform the problem, such that we can apply linear classification techniques to non-linear data. Applying the kernel equations arranges the data instances in such a way within the multi-dimensional space, that there is a hyper-plane that separates data instances of one kind from those of another. The kernel equations may be any function that transforms the linearly non-separable data in one domain into another domain where the instances become linearly separable. Kernel equations may be linear, quadratic, Gaussian, or anything else that achieves this particular purpose. Once we manage to divide the data into two distinct categories, our aim is to get the best hyper-plane to separate the two types of instances. This hyper-plane is important because it decides the target variable value for future predictions. We should decide upon a hyper-plane that maximizes the margin between the support vectors on either side of the plane that is displayed in the Figure 1.

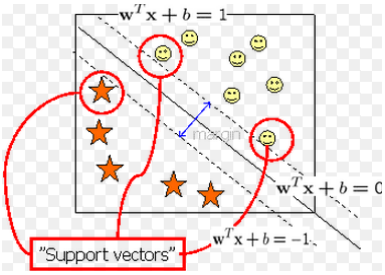

Fig. 1. Support Vector Machine

The data instances that were not linearly separable in the original domain have become linearly separable in the new domain, due to the application of a function (kernel) that transforms the position of the data points from one domain to another. This is the basic idea behind Support Vector Machines and their kernel techniques. Whenever a new instance is encountered in the original domain, the same kernel function is applied to this instance too, and its position in the new domain is found out. In our experiments too, it is seen that Support Vector Machines usually have the highest accuracy among any of the other classification methods.

Naïve Bayes Classifier:

The Naïve Bayes classifier [1] is based on the Bayes rule of conditional probability. It makes use of all the attributes contained in the data, and analyses them individually as

though they are equally important and independent of each other. For example, consider that the training data consists of various animals (for example: elephants, monkeys, and giraffes), and our classifier has to classify any new instance that it encounters. We know that elephants have attributes like they have a trunk, huge tusks, a short tail, are extremely big, etc. Monkeys are short in size, jump around a lot, and can climbing trees; whereas giraffes are tall, have a long neck and short ears.

The Naïve Bayes classifier will consider each of these attributes separately when classifying a new instance. So, when checking to see if the new instance is an elephant, the Naïve Bayes classifier will not check whether it has a trunk and has huge tusks and is large. Rather, it will separately check whether the new instance has a trunk, whether it has tusks, whether it is large, etc. It works under the assumption that one attribute works independently of the other attributes contained by the sample. In our experiments, it is seen that the Naïve Bayes classifier shows a drop in performance, when compared with K-NN and Support Vector Machines.

K-Nearest Neighbor:

The K-nearest neighbor [15] algorithm is a method for classifying objects based on closest training examples in the feature space. Unlike all the previous learning methods, K-NN doesn't build the model from the training data. No explicit model for the probability density of the classes is formed; each point is estimated locally from the surrounding points. The k-nearest-neighbor classifier is commonly based on the Euclidean distance between a test sample and the specified training samples. Given a test instance, a distance metric is computed between the test instance and all training instances, then the instance k nearest neighbors are selected from the training data as per defined in the following figure.

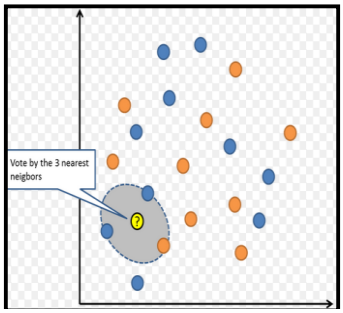

Fig. 2. 3-Nearest Neighbor

We choose SVM, because it is an immensely powerful classifier and it is more suited for 2-class problem. In addition, we compared experimentally SVM, Naïve Bayes and K-NN in performance and conclude that SVM has very good predictive power.

3.2 RapidMiner and Rapid Analytics:

The Community Edition of RapidMiner [2, 12] (formerly known as "Yale") is an open source toolkit for data mining. It provides the ability to easily define analytical steps and generate graphs. It is an environment for machine learning and data mining

experiments. RapidMiner provides a GUI which generates an XML (eXtensible Markup Language) file that defines the analytical processes the user wishes to apply to the data. This file is then read by RapidMiner to run the analyses automatically. While these are running, the GUI can also be used to interactively control and inspect running processes. RapidMiner can be used for text mining, multimedia mining, feature engineering, data stream mining and tracking drifting concepts, development of ensemble methods, and distributed data mining. RapidMiner provides data loading and transformation (ETL), data preprocessing and visualization, modeling, evaluation, and deployment. RapidMiner was rated as the fifth most used text mining software (6%) by Rexer's Annual Data Miner Survey in 2010. It is implemented in JAVA and available under GPL among other licenses. Internal XML representation ensures standardized interchange format of data mining experiments. GUI, command-line mode, and JAVA API allow invoking RapidMiner from other programs. In RapidMiner, several plugins are available for text processing, web mining etc. as well as a broad collection of data mining algorithms such as SVM, decision trees and self-organization maps.

Rapid Analytics [13] is the first open source business analytics server available. Rapid Analytics was built around the most widely used data mining solution RapidMiner and adds features like remote execution, scheduled processes, quick web service definitions, and a complete web-based report designer. Rapid Analytics is the new data mining server solution that uses RapidMiner both as a data mining engine and as a front-end to design data mining processes. We chose RapidMiner and Rapid Analytics for our implementation described in next section. First, it contains broad collection of plugins as well as large number of supervised learning methods. Second, classification engines created in RapidMiner but can be stored in remote repository to execute it remotely on the Rapid Analytic server at regular time interval.

4 Proposed Technique

In this section, we present our approach that includes (i) preparing a custom dictionary to label reviews as truthful or deceptive; (ii) the 'deceptive' classifier to predict testing data as a deceptive or truthful (iii) PHP script to label review as useful or not-useful; (iv) 'useful' classifier to predict testing data is either useful or not-useful; (v) "ranking" model to rank the reviews.

4.1 **Spam Review Detection:**

In general, spam review detection can be regarded as a classification problem with two classes, fake and non-fake. Machine learning models can be built to classify each review as deceptive or truthful. To build a classification model, we need labeled training examples of both classes. There was no labeled dataset for product opinion spam prior to this project. Recognizing whether a review is a deceptive opinion spam is extremely difficult if it has to be done manually reading the review because one can carefully craft a spam review which is just like any other genuine review. We prepared the dictionary for fake and non-fake reviews by adding knowledge from the dataset which is available on http://www.cs.uic.edu/~liub/FBS/CustomerReviewData.zip and using Cornell model. To prepare dictionary we passed reviews through Cornell model that tokenizes words based on specialized characters (like space, full stop, exclamation,

question mark etc.) in each sentence and puts it into any one of the appropriate category along with weight like high positive (+3), moderate positive (+2), low positive (+1), neutral (0), high negative (-3), moderate negative (-2) or low negative (-1). Some of words from neutral category of Cornell model are important for our domain and we placed those important words into positive or negative category with weight from http://www.cs.uic.edu/~liub/FBS/CustomerReviewData. After putting each word of each sentence into any one of six categories along with weight, we calculated final weight for each unique word based on our formula as follows:

$$\textit{Weight of each word} = \frac{WordCount * Weight}{TotalWordCount}$$

More precisely we can say that,

$$\textit{Weight of each non-fake word} = \frac{WC_{HP} * 3 + WC_{MP} * 2 + WC_{LP} * 1}{TotalWordCount_{Positive}}$$

where WC_{HP} is the count of a particular word in high positve category, WC_{MP} is the count of a particular word in medium positve category, WC_{LP} is the count of a particular word in low positve category.

$$\textit{Weight of each fake word} = \frac{WC_{HN} * -3 + WC_{MN} * -2 + WC_{LN} * -1}{TotalWordCount_{Negative}}$$

where WC_{HN} is the count of a particular word in high negative category,

WC_{MN} is the count of a particular word in medium negative category

WC_{LN} is the count of a particular word in low negative category

\Using above formula, we prepared two wordlists for fake and non-fake reviews along with their corresponding weights. We called that dictionary through a php script to label the review as fake or non-fake based on final summation of all words in each review. If final summation of weight for fake and non-fake words of a review are positive then it is labeled as "non-fake" otherwise it is labeled as "fake".

Building Models Using LibSVM

The first component of the framework is the 'deception' classifier, which predicts whether each unlabeled review is non-fake (truthful) or fake (deceptive). As mentioned previously, we labeled training review as deceptive or truthful, so that we can train 'deception' classifiers using a supervised learning algorithm. We tried three supervised learning algorithms: support vector machine (SVM), K-NN, Naive Bayes to classify product review using two pre-classified training sets: deceptive and truthful. Our work has shown that SVM trained and performs well in deception detection tasks. We found that SVM creates a hyper plane to best separate the two planes and it outperforms the other two classifiers. We trained SVM classifiers using software package of RapidMiner tool. Results of evaluation are presented in the next section.

4.2 Useful Review Detection:

In general, useful review detection can be regarded as a classification problem with two classes, useful and not-useful. Machine learning models can be built to classify each review as useful or not-useful. To build a classification model, we need labeled training examples of both useful and not-useful class. There was no labeled dataset for product opinions as useful and not-useful at the time of project (to the best of our knowledge). However, to recognize review is useful or not, we considered reader's

rating on consumer's review. Using php we labeled reviews as useful if reader's rating is greater than 40% or as a not-useful review, if reader's rating is less than 40%.

Building Model Using LibSVM

The second component of the framework is 'useful' classifier, which predicts whether each unlabeled review is Useful or Not-Useful. As mentioned above, we labeled training data, so that we can train 'useful' classifiers using a supervised learning algorithm. We tried different supervised algorithms like Naïve Bayes, K-NN, and SVM. Our work has shown that SVM trained and performs well in useful or not-useful detection tasks as compared to other algorithms. We train SVM classifiers using the software package of RapidMiner tool. Results of the evaluation are presented in the next section.

4.3 Ranking Reviews:

The last component of the framework is the 'Ranking' Model. This model takes the output from the 'deceptive' classifier and 'useful' classifier as input to rearrange the reviews based on weight (confidence) of fake, non-fake, useful, and not-useful. Higher sort priority is given to deceptive/truthful reviews and then to useful/not-useful reviews. Results of evaluation of the 'ranking' model are presented in the next section.

4.4 Implementation:

For the implementation of our approach we used RapidMiner, XAMPP, Rapid Analytics tools. We created a PHP script to collect product (e.g. camera) reviews from amazon and Epinion sites. To label training data, we prepared dictionary of words for deceptive/truthful reviews and labeled the reviews by using the dictionary in the PHP script. We created another PHP script to label training set as useful or not-useful based on reader's rating. We utilized RapidMiner tool and its supervised learning method, e.g. SVM, for building the "deceptive" classification model and "useful" classification model as well as "ranking" model. For testing purpose, we designed HTML page to enter a product review. This review is stored in a database and when the RapidMiner process is executed, it will fetch reviews from the database and based on the classifier it is processed and results (reviews with classification) are stored in the database. Using the HTML page, the result of both classifiers can be displayed.

5 Experimental Results

For evaluation, we trained both our models using different types of datasets such as balanced and imbalanced. The training dataset for 'deceptive' classifier had 1348 reviews in the imbalanced dataset and 140 reviews in the balanced dataset. The training dataset for the 'useful' classifier had 5003 reviews in the balanced dataset and 5103 in the imbalanced dataset. The following experimental result show that 'deceptive' classifier gives better performance using imbalance dataset and 'useful' classifier performs well using balanced dataset with SVM classification algorithm. We calculated the performance of our models using the following formula.

Performance, $G = \sqrt{(Sn * Sp)}$

where Sn is the sensitivity and Sp is the specificity

$Sn = \frac{TP}{TP+FN}$ and $Sp = \frac{TN}{TN+FP}$

where TP is the number of true positives

TN is the number of true negatives

FP is the number of false positives

FN is the number of false negatives

Table 1: Fake/Non-Fake Classifier Performance

Algorithm	Performance (G=sqrt(spec*sens)) Imbalance Data - 1348	Performance (G=sqrt(spec*sens)) Balance Data - 140
LinearSVM	65.58%	70%
K-NN	62.18%	64.22%
Naive Bayes	60.34%	69.98%

We observed that SVM trained and performed well in deception detection tasks. We found that SVM creates a hyper plane to best separate the two planes and it outperforms the other two classifiers with an accuracy peak at about 66%. Cross-validated classifier performance results are presented in Table 1.

We tried different supervised algorithms like Naïve Bayes, K-NN, and SVM for "Useful" classifier. Evaluation results show that SVM trained and performed well in useful or not-useful detection tasks as compared to other algorithms. This approach has been evaluated to be nearly 78% accurate at detecting useful or not-useful in a balanced dataset. Cross-validated classifier performance results are presented in Table 2. Results of the ranking model are presented in Table 3.

Table 2: Useful/Not Useful Classifier Performance

Algorithm	Performance (G=sqrt(spec*sens)) Balance Data (5003)	Performance (G=sqrt(spec*sens)) Imbalance Data (5103)
LibSVM	78.29%	70%
K-NN	77.07%	72.58%
Naive Bayes	73.79%	73.05%

Table 3: Top Ranked Reviews

	Truthful/Deceptive		Useful/Not-Useful	
Consumer Reviews For Product	Result	Confidence	Result	Confidence
bought thi camera becau light photo capabl disappoint need camera could n	deceptive	87.60%	Useful	52.60%
first love canon camera have gotten point where will onli canon camera tri	truthful	84.70%	Useful	58.00%
thi excel camera class len particular impress macro telephoto light would be	truthful	83.90%	Not Useful	52.80%
soni cybershot went dead look someth replac someth afford price compact	truthful	83.30%	Useful	84.70%
thi camera near splurg purcha part regret find that grab thi camera when le	truthful	82.30%	Useful	78.70%
bought thi camera special snorkel littl scuba photo camera great just snorke	deceptive	76.80%	Not Useful	51.30%
bought thi upgrad canon thank their menu featur similar like learn whole cai	deceptive	75.50%	Useful	69.80%
befor went bigger zoom great camera great pictur amaz video super zoom	deceptive	74.30%	Not Useful	62.60%
prior canon broke dai befor christma rush store plai with canon fell love wit	deceptive	73.50%	Useful	51.90%
have just bought thi canon powershot current stai australia price australia hi	deceptive	73.30%	Not Useful	53.40%
bought thi canon befor trip europ happi that qualiti pictur veri good even wh	deceptive	73.30%	Useful	55.70%

6 Summary and Future Work

As individuals and businesses are increasingly using reviews for their decision-making, it is critical to detect spam reviews. We presented our approach for detecting spam, not-useful reviews and prioritization of the reviews based on their weight (confidence). The evaluation shows that 'deceptive' classifier and 'useful' classifier is nearly 66% and 78% accurate respectively. Various supervised learning methods were used and we observed that SVM worked best as it is an immensely powerful classifier and it is well suited for 2-class problem. In addition, we compared experimentally SVM, Naïve Bayes and K-NN in performance and concluded that SVM has very good predictive power. Online reviews are worthless if they are not honest opinion. Our models, can give an idea to users on which reviews are non-fake and useful as well as which reviews should be completely ignored in product purchase decision-making thereby helping choose the right product. Future work might explore other methods for labeling online reviews, and will focus on improving the accuracy and more sopphisticated techniques for detecting spam reviews.

References

[1] S. B. Kotsiantis, I. Zaharakis, P. Pintelas. "Supervised machine learning: A review of classification techniques." Emerging artificial intelligence applications in computer engineering 160 (2007): 3-24.

[2] M. Hofmann, R. Klinkenberg, eds. RapidMiner: Data mining use cases and business analytics applications. CRC Press, 2013.

[3] C. Danescu-Niculescu-Mizil, G. Kossinets, J. Kleinberg, and L. Lee. How opinions are received by online communities: a case study on amazon.com helpfulness votes. In 18th international conference on World Wide Web (WWW), pp. 141-150, 2009.

[4] C. Dellarocas. Immunizing online reputation reporting systems against unfair ratings and discriminatory behavior. In ACM Conference on Electronic Commerce (EC), pp. 150-157, 2000.

[5] I. Fette, N. Sadeh-Koniecpol, A. Tomasic. Learning to Detect Phishing Emails. In Proceedings of International Conference on World Wide Web (WWW), 2007.

[6] N. Jindal and B. Liu. Opinion spam and analysis. In WSDM, 2008.

[7] S.-M. Kim, P. Pantel, T. Chklovski, M. Pennacchiotti. Automatically assessing review helpfulness. In Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 423-430, 2006.

[8] N. Jindal and B. Liu. Analyzing and Detecting Review Spam. In IEEE Intl. Conference on Data Mining (ICDM), pp. 547-552, 2007.

[9] J. Liu, Y. Cao, C.-Y. Lin, Y. Huang, and M. Zhou. Low-quality product review detection in opinion summarization. In Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL), 2007.

[10] G. Wu, D. Greene, B. Smyth, and P. Cunningham. Distortion as a validation criterion in the identification of suspicious reviews. In Proceedings of the First Workshop on Social Media Analytics (SOMA) at SIGKDD, pp. 10-13 , 2010.

[11] 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 (WWW), pp. 201-210, 2012.

[12] RapidMiner: http://www.softwarenhardware.com/tag/rapidminer-tutorial/ [Last Accessed: Mar 2018]

[13] Rapid Miner & Rapid Analytics [Online] http:// www.rapidi.com/downloads/brochures/RapidMiner_ Fact_Sheet.pdf [Last Accessed: March 2018]

[14] A-M. Popescu, O. Etzioni. Extracting Product Features and Opinions from Reviews. EMNLP'2005.

[15] T. Seidl, H. Kriegel. "Optimal multi-step k-nearest neighbor search." ACM Sigmod Record. Vol. 27. No. 2. ACM, 1998.

[16] L. Li, B. Qin, W. Ren, T> Liu. "Document representation and feature combination for deceptive spam review detection." Neurocomputing, Volume 254, 2017, pp. 33-41.