MATEC Web of Conferences **58**, 03003 (2016) DOI: 10.1051/matecconf/20165803003
BISSTECH 2015

FAKE REVIEW DETECTION FROM A PRODUCT REVIEW USING MODIFIED METHOD OF ITERATIVE COMPUTATION FRAMEWORK

Eka Dyar Wahyuni [1] and Arif Djunaidy [2]
[1]Major In Information System, Dept. Of Informatics Engineering, ITS, Surabaya, East Java, Indonesia.
University of Pembangunan Nasional "Veteran" Jawa Timur, Surabaya, East java, Indonesia.
[2]Dept. of Information Systems, ITS, Surabaya, East Java, Indonesia.
E-Mail: [1]ekadyar.w@gmail.com,[2]adjunaidy@gmail.com

ABSTRACT

The rapid growth of the Internet influenced many of our daily activities. One of the very rapid growth area is ecommerce. Generally e-commerce provide facility for customers to write reviews related with its service. The existence of these reviews can be used as a source of information. For examples, companies can use it to make design decisions of their products or services, while potential customers can use it to decide either to buy or to use a product. Unfortunately, the importance of the review is misused by certain parties who tried to create fake reviews, both aimed at raising the popularity or to discredit the product. This research aims to detect fake reviews for a product by using the text and rating property from a review. In short, the proposed system (ICF++) will measure the honesty value of a review, the trustiness value of the reviewers and the reliability value of a product. The honesty value of a review will be measured by utilizing the text mining and opinion mining techniques. The result from the experiment shows that the proposed system has a better accuracy compared with the result from iterative computation framework (ICF) method.

Keywords: fake reviews, fake reviews detection, opinion mining, sentiment analysis, text mining, ICF

INTRODUCTION

Reading product reviews before buying the product becomes a habit, especially for potential customers. If they want to buy a product, they usually read reviews from some customers about the current product. If the review is mostly positive, there is a big chance to buy the product, otherwise if it's mostly negative, they tend to buy other products. While, for a company, the positive reviews from customers can generate significant financial benefits for businesses, it can be used as input for decisions related to product design and what services are provided to customers (Heidary et al., 2015).

Related with the financial benefits gained as a result of the positive reviews about the product / service from the customer, the fraudsters tried to play the existing system by writing fake reviews and provide an assessment that is not fair to promote or discredit a product or service (Heidary et al., 2015). Individuals like this are called spammers and their activity called opinion spamming (Liu, 2012).

Automatic detection of spammers is a very important work but still lacks in research. Unlike other types of spam, such as web spam or email spam, spam on a review is far more difficult to detect. The main reason is that spammers can easily disguise themselves. Thus it is difficult, for users to recognize, while web spam or email spam, one can determine spam or not without much difficulty (Heidary et al., 2015). The other reason is difficult to find good data (gold standard) related with fake and genuine reviews to build a model, because it is very difficult, to manually identify/labeling whether the review is fake or not just by reading them (Liu, 2012).

In general, the detection of fake reviews can be seen as a classification problem with two classes, fake and genuine. This classification problems are generally resolved with supervised method. However, as explained earlier, the main problem is the difficulty to identify fake reviews manually (from reading those reviews), because spammers can create a fake review as similar as possible to the original one (Liu, 2012). Because of these difficulties, there is no reliable data related to fake and genuine reviews that can be used to conduct training. Regardless of these difficulties, some supervised algorithms have been developed and evaluated.

Due to the absence of reliable data that has been labeled fake or genuine for training, Jindal and Liu (2008) utilized duplication in the review. In their study of 5.8 million reviews and 2.14 million reviewers of amazon.com, they found a large number of duplicate and near duplicate review, which showed that the presence of fake reviews is widespread. This is likely due to the difficulty of writing some new review when the spammers never buy the product or use the service. They tend to used the same review or slightly revised for different products. This duplication can be divided into four categories, that is (1) duplicates from the same customer id on the same product, (2) duplicates from different customer id on the same product, (3) duplicates from the same customer id on different products, (4) duplicate from different customers id on different products. The first type of duplication can be generated from reviewer's mistakes in clicking the submit button several times (which can easily be checked by the date of entry of the data). However, the last three types of duplication is most likely fake review. Thus, the training data used the last three types of duplication as the fake review and the rest of the other reviews as the original one.

In line with the growing popularity of sentiment analysis and opinion mining, and the increasing number of publications related to the fake reviews detection. Sites that provide product reviews facilities are implementing a wide range of algorithms that utilize a wide range of indicators to detect spammers. But on the other hand, spammers are able to learn from their mistakes and change their tactics and eventually adapt to avoid antispam techniques that are already equipped with these indicators, so that in the end, the algorithm that utilized duplication approach no longer can be used to detect fake reviews

© The Authors, published by EDP Sciences. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (http://creativecommons.org/licenses/by/4.0/).

anymore. Motivated by the difficulties in manually labeling training data, Wang et al.,(2011) developed an unsupervised method. Computational methods were developed to measure the trustines of reviewers, the honesty of reviews, and the reliability of stores. Properties of review used in this study is its rating. They examined how the interaction between nodes (reviews, reviewers and shop) in a graph can reveal the cause of spam. They also created an algorithm called iterative computation framework (ICF) for calculating the trustiness of reviewers, the honesty of reviews, and the reliability of stores by exploring interdependencies between the three of it. This approach successfully detected fake reviews that are not duplicate with the level of precision (the ratio between the correct result of identification of spam obtained from the algorithm with the results from manually identified by humans) by 49%. The low value of the accuracy of the algorithm is likely influenced by the use of the rating property in measuring the honesty of a review, without looking at the semantic (did not utilizing the text property).

This study attempt to modify ICF by utilizing semantic aspects of a review by using opinion mining (in this research, we used sentiment polarity). Aspects extraction process needs to be done before executing opinion mining on a review. In this research, aspects that want to be axtracted are an attribute from the product. Extracted attributes are an attribute from the review that is explicitly mentioned in the review. This attributes extraction performed by using the FP-Growth. The next process is to determine the sentiment from the attributes by using opinion mining.

The fake review detection will compare the attributes and sentiments from the attributes of the reviews with the attributes and sentiments from the attributes of the other reviews and take into account the value of the similarity between the two reviews. To calculate the value of similarity, this research used Jaccard Similarity because euclidean distance is a measure of the distance, whereas the cosine similarity, the Jaccard coefficient and the Pearson coefficient is a measure of similarity (Sandhya, et al., 2008) and also because Jaccard Similarity is suitable for a wide range of applications, such as textual similarity or similarity of the buying habits of customers (Leskovec, Rajaraman, & Ullman, 2011). The end result of this detection application is the trustines value of reviewers, the honesty value of reviews, and the reliability value of a product.

Opinion mining and sentiment analysis

Textual information generally classified into two main categories, ie the fact and opinion. Facts are objective statements about entities and events in the world. Opinion are subjective statements that reflect the sentiment or perception of entities and events (Liu, 2009). Sentiment analysis, also called opinion mining is a study that analyze opinions, sentiments, evaluation, assessment, stance and emotions towards an entity such as products, services, organizations, individuals, issues and events, topics and attributes.

Opinions are very important in decision making. This applies both to individuals and organizations. The impact, the technology application of opinion mining has wide coverage. For individual consumers, if she/he want to buy the product, reads the opinion summary from the previous customer was very helpful in making a decision to buy or not. The individual can also compare the opinion summary from competing products. This is way better than reading from a various review and create the big picture of the advantages and disadvantages of a product. For organizations and businesses, opinion mining can be used to see how the costumer's views related its products and the products of its competitors. This information is not only useful for marketing and benchmarking of the products, but also can be used for design and product development (Liu, 2009).

Opinion mining starts with identifying the words that contain an opinion, ie, amazing, beautiful, ugly, etc. There are many studies that have been mine the words and identify its semantics orientation (as positive or negative). Hatzivassiloglou, V. and McKeown, K. in Liu (2009) identifies the linguistic rules that will be used for the identification of opinion words in a large corpus. Bootstrap approach carried out by Hu, M and Liu, B; Kim, S. and Hovy, E. in Liu (2009) by using a small set that contains the opinion words and then search for its synonyms and antonyms in WordNet. The major development of sentiment classification in a document level is done by Dave, D., Lawrence, A., and Pennock, D; Pang, B., Lee, L. and Vaithyanathan, S; Turney, P in Liu (2009) which classify each document reviews about an object (eg, movies, camera, or car) into positive or negative sentiment.

Association rule mining

In data mining the task of finding relationship among data is very important. Many algorithm developed for this task, ie apriori, fp growth, eclat, relim etc. Among other algorithm, apriori and fp growth have been studied in large scale in the past few years.

The Apriori algorithm was proposed by Agarwal and Srikant in 1994. Apriori uses a "bottom up" approach, where frequent subsets are extended one item at a time (a step known as candidate generation), and groups of candidates are tested against the data. Apriori uses breadthfirst search and a hash tree structure to count candidate item sets efficiently. It generates candidate item sets of length k from item sets of length k-1. Then it prunes the candidates which have an infrequent sub pattern. After that, it scans the transaction database to determine frequent item sets among the candidates. Candidate generation and support counting is expensive (in space and time).

Whereas The FP-Growth Algorithm, proposed by Han, is more efficient. This algorithm used an extended prefix-tree structure for storing compressed and crucial information about frequent patterns named frequent-pattern tree (FP-tree). Then it extract frequent itemsets directly from the FP tree. In his study, Han proved that his method outperforms other popular methods for mining frequent patterns, e.g. the Apriori Algorithm and the TreeProjection (Han, et al., 2004). Unfortunately, this task is computationally expensive, especially when a large number of patterns exist.

Jaccard simmilarity

Jaccard Similarity is often called Jaccard Coefficient or Tanimoto Coefficient, measured similarity as an intersection of an object that is divided by union of the object (Sandhya, et al, 2008; Leskovec, Rajaraman, & Ullman, 2011). Jaccard Similarity of the set S and T are JS (S, T), with the following equation

$$JS(S,T) = \frac{|S \cap T|}{|S \cup T|} \quad (1)$$

METHODOLOGY

Data preprocessing

Dataset used in this research was dataset from another research conducted by McAuley and Leskovec (2013), which was a product review from Amazon.com from June 1995 - March 2013, data retrieved from https://snap.stanford.edu/data/web-Amazon.html.
Examples of data as shown in the figure-1

```
product/productId: B000FTPOMK
product/title: 14k Yellow Gold Butterfly Pendant, 16"
product/price: unknown
review/userId: A3T4KKUM1JATUX
review/profileName: Disappointed Sony customer
review/helpfulness: 1/1
review/score: 4.0
review/time: 1178150400
review/summary: pretty necklace
review/text: It is a nice made necklace, and the butterfly pendant looks beautiful. I love it.
```

Figure-1. Product review

Product / productId : an identification number (standard Amazon).
product / title : describes the product name.
product / price : the price of the product.
review / userId : user identification number.
review / profileName : user name.
review / helpfulness : the number of users who assess whether this review quite helpful.
review / score : the rating of the product.
review / time : the time when the review submitted (unix time).
review / summary : a summary from reviews and
review / text : is the content of the review.

In the preprocessing stage, the data parsed token by token and put on the review tables. Tokens can be any word or character. For this process the applications developed using PHP and AJAX. Once the application executed, the database will be filled, but the existing data still can not be used for the next process, its still need to be filtering. In this research, filtering (in the database) are held in 4 stages. The first stage is the removal of duplicate review. The second stage is the removal of 'anonymous' and 'unknown' users. The third stage is the removal of unpopular product, ie a product that has less than 3 reviews. The fourth stage is the removal of inactive users, ie users that has less than 3 review. After the filtering stage finished, the data generated is clean and are ready to be analyzed.

ICF ++ method

ICF ++ method took the name of a product, the name of a reviewer, the content of a review, the number of iteration and minsup as parameter. After taking those value, this method intialize the counter to 0, and honesty value and reliability value to 1. After that, the next process are pos tagging, creation of transaction file, FP Growth, polarity generation, calculation of agreement value. The next step is iteration. Inside in an iteration, the method continues by calculating the honesty value, calculating the trustiness value, calculating the reliability value, updating agreement value using updated honesty value, updating reliability value and increement counter. Iteration in this research is 3, because the algorithm converges quite fast in practice. The process are described in the figure-2.

DOI: 10.1051/matecconf/20165803003

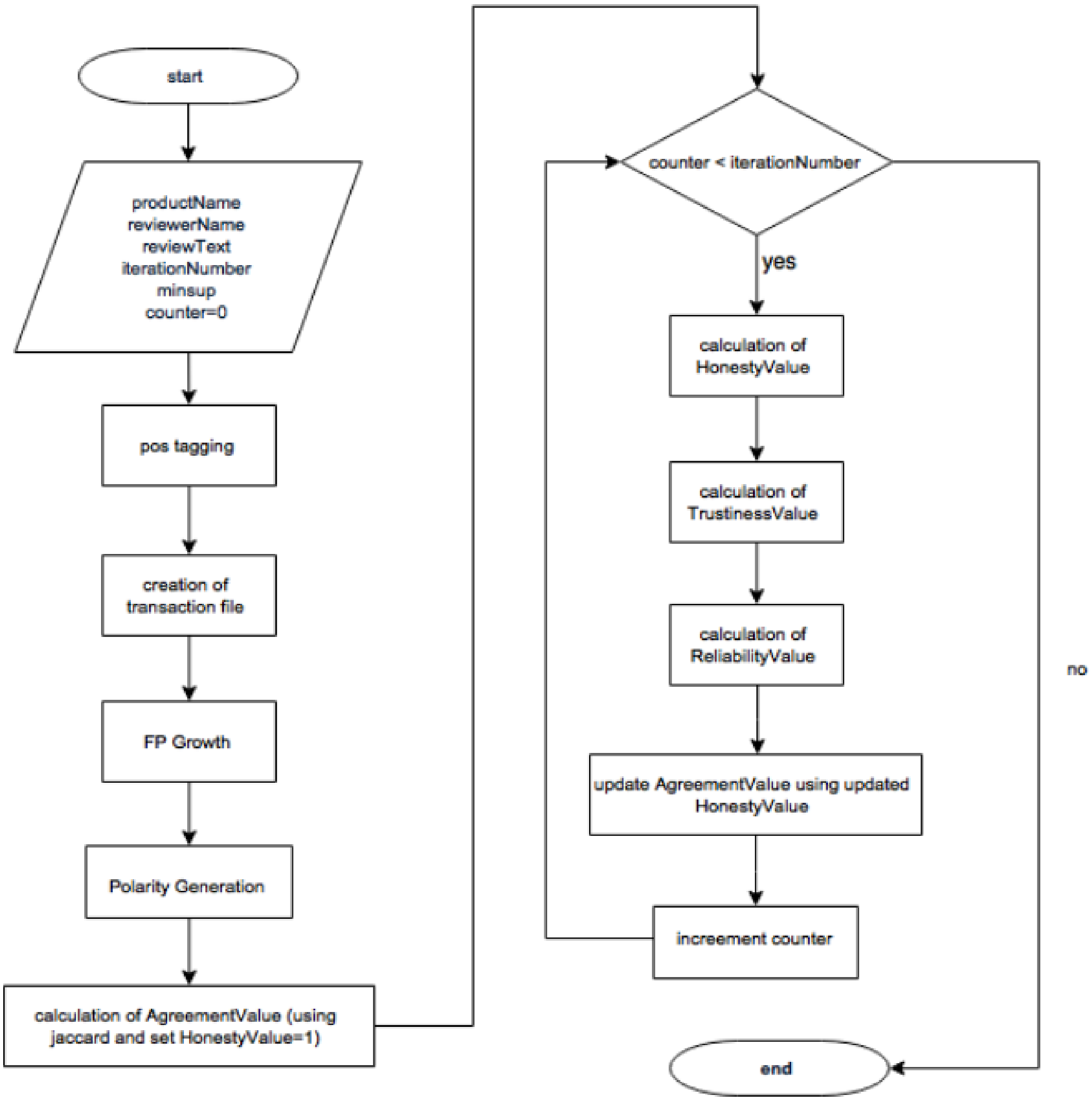

Figure-2. ICF++ flowchart

POS tagging

POS tagging was conducted by utilizing the Stanford Part-Of-Speech (POS) tagger from stanford-corenlp-3.5.1. . Stanford Pos Tagger is a software that reads text and determine the part of speech for each token. This process produced a tag in the form of abbreviations, for english tagger, the resulted abbreviations used Penn Treebank standard.

For this process, a desktop application was developed using Java. This process started with training the tagger model. Input from this process is a sentence from review and produce POS tag for each token that is stored into the database.

Creation of transaction file

Transaction file in this research contains all tokens that are stored in the database from a products that it's tag value is noun, either NN, NNP, NNPS or NNS. Each row in the transaction file was a noun from every sentence for a product. This transaction file became the input for the FPGrowth process.

FPGrowth

This process aimed to identify the features that has the most comment. In this study, the features are properties or attributes from a product. For example, for camera, the feature can be its battery, memory card etc. This information actually available on amazon, but, datasets related to this information were not publicly available, so to get information about product's feature, this research used fp-growth algorithm from association rule mining technique. This study used spmf library developed by (Fournier-Viger, et al., 2014).

The minsup used in this research was 0,1 and the confidence was 0,8. The minsup and confidence values ranging from 0 to 1. The minsup (minimum support) used for the comparison against the support that was owned by an item that compose a rule. Rule produced by this algorithm should consist of items that it's support value are exceed the minsup value. Support value was calculated from the number of item appearance in a transaction. For example, there was an item that appears two times in 5 transactions, then the support was 2/5 which is 0.4. If the minsup value entered was 0.1, then this item would appear in the output, but if the minsup value entered is 0.5 then this item will not appear in the output.

The FP Growth process in this research described as follow, input for this process are the transaction file that contains nouns. The next stage, are FP-Tree generation. FP tree is data structure that contains specific information about item, number of occurance of the item and link to other node. The next stage, this fp tree will be scanned to generate frequent pattern in the form of rule. The resulting rule will be broken down to obtain each phrase. Each of these phrases is an attribute of a product. The last stage is, the attribute will be saved to the transaction file and to the database.

Polarity generation

The feature or attribute resulted from the previous process will be identified its opinion orientation. For example, for a camera, the feature or attribute that can be identified is the battery, the comment related to the battery from userId X is "this battery drain pretty fast, and you do not want to be stuck somewhere with a dead battery". The purpose of this process is to identify the opinion orientation from a sentence that contains the attributes that were identified from the previous process. Whether its orientation positive, negative or neutral. From the example above, its opinion orientation is negative.

Sentiment prediction/opinion orientation prediction can use variety of techniques, from the words level, sentence level and others. According to Socher, et al., (2013), sentiment prediction only from words often misses, because the order of each word was ignored. As in the illustration below, the words funny and witty each are positively oriented, but in the sentence "This movie was actually neither that funny, nor super witty", causing the orientation of the overall sentence to be negative.

The input for this process are two parameters, ie a list of all tokens from a review of a current product (sentences that have been POS tagged and stored into the database), named list i. The second parameter is a list of all the features from a current product, named list f, which is obtained from the database. This process will check whether index i-th from list i contains a token from list f. If this condition is met, it will look for the orientation by using stanford.nlp.sentiment from stanford-corenlp-3.5.1. library. The output of this process is a list of features contained in a single sentence along with the orientation of the feature, as the following examples "necklace pos pendant neg".

Calculation of agreement value

The next process was calculate the agreement value that illustrates how similar the feature and its opinion orientation resulted from the polarity generation process with the other reviews. This process uses Jaccard similarity. Results from Jaccard similarity was in the range of (0,1), 0 means that the second review didn't have the same opinion orientation for any feature, 1 means both reviews had the same opinion orientation for all the features. In this research, both of the reviews agreed each other if their jaccard value exceed 0.8.

The surrounding set of review v (S_v) was the set of v's surrounding reviews about a product.

$$S_v = S_{v,a} \cap \ S_{v,d} \tag{2}$$

where $S_{v,a}$ is set of the reviews that agree with each other, $S_{v,d}$ is a set of reviews that disagree with each other. From the definition above, $S_{v,a}$ is in the form of a list of a reviewID that contains the other review which it's jaccard value exceed 0.8. Whereas, $S_{v,d}$ contains other review which it's jaccard value less than 0.8.

The agreement value from a review v within a period Δt is described in the following equation

$$A_n(v, \Delta t) = \frac{2}{1 + e^{-A(v,\Delta t)}} \tag{3}$$

$$A(v, \Delta t) = \sum_{i \in S_{v,a}} \ T(\kappa_i) - \sum_{j \in S_{v,d}} \ T(\kappa_j) \tag{4}$$

where A is the agreement value of the target review. In this research, we used 3 months before and after the posting time of the target review as the time window Δt.

Calculation of honesty value

Review's honesty value is described in the following equation

$$H(v) = |\, R(\Gamma_v) | A_n(v, \Delta t) \tag{5}$$

Where $R(\Gamma_v)$ is product's reliability value that is being reviewed.

Calculation of trustiness value

Reviewer's trustiness value is defined in the following equation

$$T(r) = \frac{2}{1 + e^{-H_r}} - 1 \tag{6}$$

Where T (r) is the reviewer's trustiness value.

Calculation of reliability value

Product's reliability value (R (p)) is described in the following equation

$$R(p) = \frac{2}{1 + e^{-\theta}} - 1 \tag{7}$$

Where

$$\theta = \sum_{v \in U_s, T(\kappa_v) > 0} \ T(\kappa_v)(\Psi_v - \mu) \tag{8}$$

Where Ψ_v is the rating from review v, while is the median value of the rating, which is 3, in the range of 1-5 rating.

RESULTS AND DISCUSSIONS

There are two types of tests in this research, which is verification of results test and performance test. The purpose of the test of verification of the results is to check whether the calculations performed by the application and manualy calculated has the same process and the same result. This test also intend to check if the given input has the correct output. While the aim of the performance test is to compare results from the ICF ++ method with another fake review detection methods. Method from Wang, et al., (2011) is reimplemented using java to see the performance. Then the fake review detection methods in Wang's research used as a reference comparison with method ICF ++.

Result from verification test

Applications that has developed take 4 input, ie, product name, reviewer name, review (text) and minsup. The application will perform three iterations. Each iteration will be shown in the system and then compared with the results from manual calculations which also iterate 3 times. Summary of the results of the test can be seen in the following table.

Table-1. Summary result of manual calculation and ICF++

no	process	Manualresults			ICF++ results			conclusion
		I 1	I 2	I 3	I 1	I 2	I 3	
1	agreement	0	-1,227	-1,6531	0	-1,227	-1,6531	**verified**
2	agreementVal	0	-0,5466	-0,6786	0	-0,5466	-0,6786	**verified**
3	honesty	0	-0,4163	-0,5168	0	-0,4163	-0,5168	**verified**
4	sumHonesty	-0,4621	-1,5615	-1,8197	-0,4621	-1,5615	-1,8197	**verified**
5	trustiness	-0,227	-0,6531	-0,7211	-0,227	-0,6531	-0,7211	**verified**
6	reliability	-0,7616	-0,7616	-0,7616	-0,7616	-0,7616	-0,7616	**verified**
7	theta	-2	-2	-2	-2	-2	-2	**verified**

From table 1, it can be concluded that for every process performed by the application, all the process and result are verified, because for a given input, the output generated by applications has been in line with the expected result, which is derived from manual calculations.

Result from performance test

The Application calculated the honesty value of a review, the trustiness value of the reviewers and the reliability value of a product. The algorithm developed by Wang, et al., (2011) will converges very fast. Therefore, from 855928 review data, not all of it being processed with these two models. Systematic sampling technique carried out, that will select elements number n-th as a sample. In this study, the interval used (n) is 800, so every reviews multiple of 800 taken as sample. The number of samples to be tested for both models is 1070. The steps of this test is described as follows

1. Fake review detection by ICF
2. Fake review detection by ICF ++

Result evaluation

This research used IR-based evaluation strategy (Wang, et al., 2011) with the following steps

1. both models identified fake reviews.
2. human evaluator decide whether the review fake or genuine, in order to calculate the precision values which illustrates the performance of the model.

A method is said to be effective, if different juries have the same assessment related a current review, so in this research, consistency of assessment used as one of the evaluation criteria. The evaluator used in this study were 3 graduates student from computer science and has considerable experience in online shopping.

Judging suspicious spammer was a complex task, which requires intuition and also ability to search for additional information. To decide whether this reviewer is a review spammer or not, the judge must read the reviews and collect evidence about relationship between these reviews to the other reviews, also how the relationship between reviews of all products that has reviewed by this reviewer with another review by other reviewers. To standardize the judgement process, the three judges agreed on two conditions as our evidence to claim that a reviewer is a potential spammer.

1. A reviewer is suspicious if she/he has a significant number of reviews that gives opposite opinion to other reviews about the same product.
2. A reviewer is suspicious if she/he has a significant number of reviews that gives opposite opinion to other reviews about some product as compared to evidences presented by general web search results (eg, the google search results).

This research took 100 suspicious reviews. In evaluation stage, if more than one judges identify that the targetted review was fake, it was labeled as fake. To study the agreement between each jury, this research used Fleiss kappa (Wang, et al., 2011), which is an inter-evaluator agreement measure for any number of evaluators. Kappa values for each model are summarized in the following table.

Table-2. Kappa values of ICF and ICF++

Model	Kappa Value	agreement Interpretation
ICF	0.648823	substantial agreement
ICF ++	0.697581	substantial agreement

From table 2 can be concluded that the agreement of the three judges for all methods is more than 0.6 and less than 0, 8 which represents substantial agreement.

Precision values of the model ICF is summarized in the following table.

Table-3. Precision value of ICF

evaluator	1	2	3
1	54	46	46
2		57	49
3			56

For the ICF method, our judges identified 57 suspicious reviews, so the precision is 57%. Table 3 shows the agreement of evaluator. Evaluator 2 identified 57 suspicious reviews, out of which 49 were recognized by Evaluator 3 and 46 were caught by Evaluator 1.

Precision values of the model ICF++ is summarized in table 4.

Table-4. Precision value of ICF++

evaluator	1	2	3
	54	51	48
2		63	55
3			59

For ICF++ method, three judges managed to identify 63 out of 100 reviews were suspicious, so the precision is 63%. Table 4 shows the agreement of evaluator. Evaluator 2 identified 63 suspicious reviews, out of which 55 were recognized by Evaluator 3 and 51 were caught by Evaluator 1.

CONCLUSION

From Table 3 and 4 can be seen that the precision value of ICF ++ method is higher than ICF method. It can be concluded that the use of ratings alone to assess whether the review is fake or genuine is inadequate, because the information that can be processed is very limited. By using text properties, as implemented in the ICF ++ method proven to improve the precision value by 6%.

The drawback of this method is, some process need to be optimized, so it can detect a fake review in a short amount of time.

REFERENCES

Coenen, F., Goulbourne, G. & Leng, P., 2003. Tree Structures for Mining association Rules. Journal of Data Mining and Knowledge Discovery, Vol 8, No 1, pp.25-51.

Forman, E.H., 1993. Facts and fictions about the analytic hierarchy process. Mathematical and Computer Modelling, Volume 17, Issues 4–5, pp.19-26.

Fournier-Viger, P., Gomariz, Gueniche, T., A., Soltani, A., Wu., C., Tseng, V. S. 2014. SPMF: a Java Open-Source Pattern Mining Library. Journal of Machine Learning Research (JMLR) 15, pp. 3389-3393.

Han, J., Pei, J., Yin, Y. & Mao, R., 2004. Mining Frequent Patterns without Candidate Generation: A Frequent-Pattern Tree Approach. Data Mining and Knowledge Discovery 8, pp.53–87.

Heydari, A., Tavakoli, M.A., Salim, N., & Heydari, Z., 2015. Detection of review spam: A survey. Expert Systems with Applications, 42(7), pp.3634–42.

Jindal, N., & Liu, B. 2008. Opinion Spam and Analysis. Proceedings WSDM '08 Proceedings of the 2008 International Conference on Web Search and Data Mining,219-230.

Khan, K., Baharudin, B., Khan, A., & Ullah, A. , 2014. Mining opinion components from unstructured reviews: A review. Journal of King Saud University - Computer and Information Sciences, 26(3), pp.258–75.

Leskovec, J., Rajaraman, A. & Ullman, J.D., 2011. Mining of Massive Data Sets. Cambridge University Press.

Liu, B., 2009. Opinion Mining. Encyclopedia of Database System, pp.1986-90.

Liu, B., 2012. Sentiment Analysis and Opinion Mining. Morgan & Claypool Publishers.

McAuley, J., & Leskovec, J. 2013. Hidden Factors and Hidden Topics : Understanding Rating Dimensions with Review Text. Proceeding RecSys '13 Proceeding of the 7th ACM conference on Recommender Systems , 165-172.

Pang, B. & Lee, L., 2008. Opinion Mining and Sentiment Analysis. Foundations and Trends in Information Retrieval Vol. 2, No 1-2, pp.1-135.

Ravi, K. & Ravi, V., 2015. A survey on opinion mining and sentiment analysis: Tasks, approaches and applications. Knowledge-Based Systems, 89, pp.14-46.

Sandhya, N., Lalitha, Y.S., Govardhan, A. & Anuradha, K., 2008. Analysis of Similarity Measures for Text Clustering. CSC Journals 2.

Savage, D., Zhang, X., Yu, X., Chou, P., Wang, Q., 2015. Detection of opinion spam based on anomalous rating deviation. Expert Systems with Applications, 42(22), pp.8650–57.

Socher, R., Perelygin, A., Wu, J.Y., Chuang, J., Manning, C.D., Ng, A.Y., Potts, C., 2013. Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank. Proceedings of the conference on empirical methods in natural language processing (EMNLP). Vol. 1631 , 1642.

Wang, G., Xie, S., Liu, B., & Yu, P.S 2011. Review Graph Based Online Store Review Spammer Detection. Proceeding ICDM '11 Proceedings of the 2011 IEEE 11th International Conference on Data Mining , 1242-1247.