Spam Review Detection through Lexical Chain Based Semantic Similarity Algorithm (LCBSS) for Negative Reviews

Rupesh Kumar Dewang [#1], Anil Kumar Singh [*2]
[#] Department of Computer Science and Engineering, Design Center,
F-12 Motilal Nehru National Institute of Technology Allahabad, Uttar Pradesh INDIA 211004
[1] rupeshdewang@mnnit.ac.in
[*] Department of Computer Science and Engineering, Computer Center,
Motilal Nehru National Institute of Technology Allahabad, Uttar Pradesh INDIA
[2] ak@ mnnit.ac.in

Abstract—**The negative spam reviews are more harmful for hotel services because the impacts of negative information are faster and greater, than positive information. The sentiment analysis for detection of spam review is not effective in today scenario because for making new spam review, the spammers instead of copying exact texts; they are combining two or more text which contains the same sentiment. The spammer used synonyms, morphological words and also shuffled some word to create new negative spam review. In this paper, we have proposed Lexical Chain Based Semantic Similarity (LCBSS) algorithm which gives better accuracy, when compared with the Bag-of-Word (BOW) model and baseline-[1] method. The Proposed LCBSS algorithm has generated feature vector which is used as input to ten supervised algorithm. Amongst ten supervised algorithm, Support Vector Machine (SVM) gave 99.75% accuracy which is the highest accuracy up-till now.**

Keyword- Review Spam, Semantic Analysis, Supervised Algorithms, Lexical Chain and Bag-of-Word.

I. INTRODUCTION

Online booking of hotels are increasing because; customers can compare the price at different websites of the same hotel. The hotel selection not only depends upon, what they are offering? This also depends upon the visited user's written feedback in the form of reviews. Reviews play very important role for users whether to book or not to book hotels. The reviews contain only positive and negative sentiments. Positive review enforces to buy service; whereas negative review shows very clear intention not to buy services. A negative review mostly affects the brand name of hotels as well as it also affects hotel chains (other area location). If, these negative reviews found fake; it will damage overall business and reputation of hotel [2]. Few papers have done their work on negative fake review [1], [3]. However, most of the researchers have done their works on positive fake reviews [2], [4]-[7]. Many researchers have showed that, the bias negative reviews have a harmful impact on product sales then the positive reviews [2], [8]. The bad feedbacks had greatest impact than the good ones. The bad information are more easily processed and considered than good [9]. Peoples are losing their trust on posted reviews due to bias negative reviews [10]. Some competitor hotels are playing the fake review posting game to take an advantage as in financial terms and also damage the hotels established reputation [3]. According to the Banerjee et al., the hotels could also involve in posting bias negative reviews on their competitor hotels, rather than to write bias positive reviews for their own hotels [8]. So, here we have continued the research on bias negative review.

The semantic based similarity used by Sandulescu et al. first time in 2015. Where, he noticed very useful assumption, which is acquired from Ott et al. 2011, the review spammers have an insufficient creativity and insufficient knowledge to write completely new detail in every new review. So, they shuffled some words with their semantic meaning. Sandulescu et al. have worked only on the semantic similarity on synonym words and used Bag-of-Word (BOW) model. The main deficiency of BOW model requires a larger space to store the words. The BOW model selects the raw words to store. These raw words selects as a features does not contain any other information holding with sentence in text.

In this paper, we have proposed a LCBSS algorithm which uses the lexical cohesion based similarity measure defined in [11]. Here, the proposed LCBSS algorithm is consist of three algorithms, first, for pre-processing of review dataset, second for construction of lexical chains described in [11] and third algorithm for TF-IRF (Term Frequency- Inverse Review Frequency) feature vector generation using selected chains.

Rest of this paper is presented as follows: In section II, we have discussed related works; in Section III, we have discussed proposed methodology; in Section IV, showed the experiment setup and results discussion and in last Section V we provided the conclusion.

II. RELATED WORK

Spam is basically defined as a distorted message sent using electronic media to numerous recipients. The messages may also contain some secret script to perform a cyber-attack by cyber criminals. The work in spam detection started with e-mails and then similarly in numerous other media for example, chat spam, newsgroup spam, web spam, blog spam, SMS spam, Internet forum spam, social spam, file sharing spam and review spam.

In the year 2008, the existence of review spam defined by Jindal & Liu [2]. According to author, review spam is new type of spam which lies on E-commerce website in the form of reviews. They defined three types of review spam:

- **Type 1 (untruthful opinions)**: when review is not given as a product alike, means review is given for promoting and demoting the product reputation by positive and negative posting reviews.
- **Type 2 (reviews on brands only)**: When the review is written in consideration of only brands name, instance of product of that brands, spammers want to damage the reputation of brands.
- **Type 3 (non-reviews)**: when the review only contains advertisement and other irrelevant opinion of the products like question and answering and random text.

Based on above categorization, authors have detected, whether review is spam or not. Another work proposed by Siddu P. Algur et al. They proposed "conceptual level similarity measure" method. This method is based on written reviews whereas reviews include product features (Pros and Cons of products defined by the reviewer) only. The authors have defined features based threshold value to classify the review as a spam or non-spam. In my opinion, authors have used only pros and cons of review, the accuracy can be increased by including complete content of review [12].

The negative and positive review impacts for decided suspicious activity of reviewer defined by Jindal et al. in 2010. Authors have proposed a general framework to finding "unexpected rules patterns" from the review dataset. They have defined expectation and unexpectedness definition on the basis of length (short relation or long relation) of the rules. In my opinion, content of the review can be played good to role to increased spam detection accuracy [13].

In earlier papers, the research in reviews spam detection based on the duplicate and near duplicate of reviews posted by the reviewers. Lau et al. in 2011 first time used the semantic concept to detecting the reviews spam. They have proposed detailed general system architecture for review spam detection. They also proposed high-order concept association-mining module for detecting words changed with their semantic meaning. Suppose, if reviewers change the semantic meaning of words for example "love" is replace with "like" then used model can detect this replacement. The proposed model cannot be applied where only a single spam review present. This model is able to work only when spammer replace few words of review with other review. Authors have given example in which "Fabulous" replace with "Fantastic" other content of review is same. In this example, if spammers replace many words meaning of review content with their semantic meaning of words then this model will be failed to detect such changes [14].

In 2013, ott et al. used the negative review dataset because negative review is pre-planned misrepresent of user thoughts. They have used as n-gram model and psycholinguistic features for analysis purpose. It used possible collaboration of sentiment and deceptive text. The major limitation of sentiment analysis based model is that, if spammer replaces the some of words with their semantic meaning than sentiment analysis model will not able to detect review is spam or non-spam [1].

Sandulescu and Ester, has identified single spam review [14] limitation and proposed model for "singleton spammer" which is based on semantic similarity between words and reviews level. They have exploited the synonymy relations between words by knowledge-based semantic similarity measure. The synonymy relation is based on WordNet synonyms database [15]. Authors have proposed second model which is used the similarity of the undisclosed topic distributions of reviews to classify them as non-spam or spam. They used Bag-of-Words (BOW) (based on Part-of- Speech) and bag-of-opinion-phrases model. The major limitation of BOW model is that, Words in BOW model not comes according to their semantic meaning.

Earlier model is not able to work when the spammer replace many of words of review with their semantic meaning. In this paper, we have proposed LCBSS algorithm which is based on words semantic meaning relations.

III. PROPOSED METHODOLOGY

In this section, we have given the detail of used dataset, proposed LCBSS algorithm and BOW model.

A. Dataset

The availability of label dataset is major issue in review spam detection. The label dataset problem is one of the major research limitations in review spam detection. Researchers depend upon the labelling of the dataset by human experts, but this method takes time to label a large number of reviews [16]. In this paper, we have used publicly available dataset [17] build by Ott et al. in 2011. In which the total number of 20 hotels are present in reviews. Here, we used 800 negative reviews, in which 400 are non-fake reviews and other 400 are fake reviews.

B. Proposed Lexical Chain Based Semantic Similarity(LCBSS) Algorithm

The proposed LCBSS algorithm has utilized a part of lexical cohesion likeness. Lexical cohesion is essentially utilized as a part of the lexical content and foundation learning of substance. In this subsection, we discussed the LCBSS algorithm. First, we explained the Pre-processing of review dataset, Construction of Chain, Features vector calculation using Term Frequency- Inverse Review Frequency (TF-IRF), Example of lexical chain construction and finally example of TF-IRF feature vector generation.

- Pre-processing of Review Dataset

In algorithm-1, firstly, we have joined all spam and non-spam reviews to build the single document. In second steps, we applied the pre-processing steps, consequently in next steps; we performed removal of stop words, tokenization and part of speech tagging. The spammer mostly tries to shuffle noun, adjective, verb, adverb and interjection with its semantic meaning. So, in step-6, we have selected these words and performed next steps.

Algorithm-1 Pre-processing for Reviews Dataset

```
1. Combine all the reviews to make a single review document.
2. Select review document for pre-processing steps.
3. For Review Document (RD) do
4. Perform tokenization.
5. Removing all stop-words.
6. Select noun, adjective, verb, adverb and interjection.
7. Perform WSD through Roget-Thesaurus and generate all thesaurus relations and choose a set of them.
8. Set and store noun, adjective, verb, adverb and interjection words as candidate words ($CW_j$) where $1 \geq j \geq n$ and filtered other words.
9. End For
```

In next step, we performed Word Sense Disambiguation (WSD). Here, the goal of WSD is to automatically disambiguate the meaning of the word from its context, i.e., identifying the sense in which a word is used and assigning that sense to it. We used dictionary based WSD methods to disambiguate the word meaning with a reference to dictionary. The dictionary is based on a thesaurus [18]. For example: "*He has taken vicks action 500 for cold," the disease sense is purporting, in "It is cold and rainy here to-day," the temperature horripilation sense is meant, while "It's too cold yesterday, only 1 degrees"*, implies the environmental condition sense.

In Rogets Thesaurus, the words sense is already stored in the name as a head file which is represented by the numbers. There are total 1044 heads files, where all sense of words is mentioned. In next step, we have stored noun, adjective, verb, adverb and interjection. In last step, we set all stored words as candidate words CW_j.

- Construction of Chains

The lexical construction depends upon the first algorithm, in which, we have marked words as candidate words (CW). These candidate words (CW) is used in algorithm-2 to generate the lexical chains. The Jayarajan et al. had been used global set for calculating the lexical chain across all documents [22]. The global set is a set of lexical chains which appeared in the document.

In algorithm-2, we have used the global set concept. Here, the global set (G) represents the set of all generated lexical chains (LC) such that, $G=(LC_1,LC_2,LC_3,\ldots\ldots.LC_n)$. The lexical chain construction in this paper is based on Morris and Horst et al., and Silber and McCoy et al. algorithms [11], [19-20].

Algorithm 2 Construction of Lexical Chains

```
    Input: Candidate Words ($CW_j$) from algorithm-1.
    Output: Lexical Chain ($LC_n$) and Global Set (G).
1.  Maintain a Global set (G) of Lexical Chains (LC), where Global set, G= {$LC_1$ , $LC_2$ ,$LC_3$,…….. $LC_n$}
    and initialized, Global set (G) = Null, Review Document RD={ $R_1$,$R_2$,$R_3$……….. $R_n$}.
2.  For each selected Review Documents $RD_i$, where $1 \le i \le n$ from algorithm-1 do
3.     Select the set of candidate words $CW_j$ where $1 \le j \le n$ from algorithm-1
4.     For each Candidate Word ($CW_j$ ) in Reviews Document RDi do
5.       If Candidate Word ($CW_j$ ) have a repetition of same word relation or inclusion of same paragraph
6.       then
7.       Store the Candidate Word ($CW_j$ ) into respective chain.
8.        Else
9.          If Candidate Word ($CW_j$ ) is not satisfying the step 5 condition and No ($LC_n$) chain is construct
10.         then
11.         Create a new chain ($LC_n$) for this Candidate Word ($CW_j$ )and Insert Lexical Chain ($LC_n$) in
            global set G, where new value of G={$LC_n$}
12.         Else go to at step 5
13.         End If
14.       End If
15.    Insert the Candidate Word ($CW_j$) to the constructed chain or created new chains in Global set G.
16.    End For
17. End For
```

Halliday and Hasan et al. have defined the lexical chain is successions of semantically related words interconnected by means of semantic relations. They build up a semantic availability between two end concepts. Lexical chains have been developed on the basis of information that contains concepts and the relations between concepts. Lexical chain captures the lexical cohesive structure of text [21]. Lexical cohesion is the cohesion which is arises from the semantic relationships between words.

The five Parameter to determine the lexical cohesion:

(1) Reiteration with identity of reference.

(2) Reiteration without identity of reference.

(3) Reiteration by means of super ordinate.

(4) Systematic semantic relation, and

(5) Non-systematic semantic relation.

The initial three relations include emphasis which incorporates redundancy of the same word in the same sense (e.g., auto and auto), the utilization of an equivalent word for a word (e.g., auto and car) and the utilization of hypernyms (or hyponyms) for a word (e.g., auto and vehicle) individually.

The last two relations include collocations i.e., semantic connections between words that regularly co-happen (e.g., football and foul). A methodical relationship is an all-out relationship. The samples of a precise relationship are words, for example, one, two, three or red, green, blue. A non-orderly relationship is one in which words co-happen because of their lexical setting, for example, auto, tire, motor are non-methodical connection.

Morris and Hirst et al. identified five types of thesaurus relations that suggest the inclusion of a candidate word in a chain.

The five thesaurus relations used are:

1. Inclusion in the same Head.

2. Inclusion in two different Heads linked by a Cross reference.

3. Inclusion in References of the same Index Entry.

4. Inclusion in the same Head Group.

5. Inclusion in two different Heads linked to a common third Head by a Cross-reference.

Morris and Hirst state that although these five relations (parameter) have defined the first two are by far the most prevalent, constituting over 90% of the lexical relationships [11].

The two relations used for the implementation of lexical chains using the ELKB are:

1. Repetition of the same word, for example: *stay, stay*.

2. Inclusion in the same Paragraph (paragraph stores in head files).

Most of the previous lexical chains algorithms used the noun for construction of lexical chain. The main reason is that, noun helps to represents better topics in the documents. In this paper, the detection of reviews spam is very complex task. Only from noun we cannot easily identify the spam reviews. So, we have considered noun, adjective, verb, adverb and interjection for this purpose. Here, we used the Rogets Thesaurus, Electronic Lexical Knowledge Base (ELKB) tool for lexical chain generation [18]. The Rogets allows building lexical chains using noun, adjective, verb, adverb and interjection. For the 800 reviews dataset (spam and non-spam reviews) total 216 lexical chains generated through algorithm-1 and algorithm-2. Fig.3, shows the generated numbers of sense for 216 lexical chains.

- Term Frequency- Inverse Review Frequency (TF-IRF) for feature vector calculation

Here, we have calculated the term frequency (tf) with respect to reviews. We defined name of algorithm as Term Frequency- Inverse Review Frequency (TF-IRF). The classical TF-IDF (Term Frequency-Inverse Document Frequency) scheme is used to check the reasons that some words occur more repeatedly which are common in reviews. In this paper, we have modified the classical TF-IDF concept in-terms of lexical chain. The TF-IRF algorithm is inspired by Jayarajan et al. They used this scheme in first time for documents clustering [22]. In present paper, the total numbers of negative reviews are 800 and total numbers of lexical chain generated are 216, so the size of Feature Vector (FV) matrix is 800×216. This generated Feature Vector (FV) matrix is used as input of the classifier.

Algorithm-3 TF-IRF Generation through Selected Lexical Chains.

```
Input: Review Documents $RD_i$, Lexical Chain $LC_n$.
  Output: Features Vector (FV) of all reviews.
  1. For selected Review Documents $RD_i$, where $1 \le i \le n$ from algorithm-1 do
  2.    Initialize Feature Vector (FV)=zero.
  3.    Select the Lexical Chain ($LC_n$ ) from algorithm-2.
  4.    Compute the term frequency (tf) of Lexical Chain $LC_n$ as:
  5.    For each constructed Lexical Chain $LC_n$, is the number of times words f of Lexical Chain $LC_n$
        present in the document.
  6.       Calculate: $tf(LC_n)=\log [ 1+ f( LC_n)]$
  7.       review frequency (rf) of a Lexical Chain $LC_n$ is the numbers of reviews in which $LC_n$ appears.
  8.       Calculate the inverse review frequency (irf) is the inverse of rf.
  9.       Calculate $irf(L)= 1/\log( 1+ rf( L))$
  10.      The final weight value of Lexical Chain $LC_n$ is $tf.irf = tf ( LC_n)\times irf( LC_n)$
  11.   End for
  12. End for
```

- Example of Lexical chain construction

We have given one example of sample of spam and non-spam reviews from the dataset [17]. Here, first pre-processing algorithm is applied then lexical chain generation algorithm is applied.

Spam: *"My **husband** and I **stayed** for two **nights** at the Hilton Chicago, and **enjoyed** every **minute** of it! The **bedrooms** are immaculate, and the linens are very soft. We also **appreciated** the free wifi, as we could **stay** in **touch** with friends while **staying** in Chicago. The **bathroom** was quite spacious, and I **loved** the **smell** of the shampoo they provided-not like most **hotel** shampoos. Their **service** was amazing, and we **absolutely loved** the **beautiful** indoor pool. I would **recommend staying** here to anyone".*

Non-Spam: *"We **stay** at Hilton for 4 **night**'s last march. It was a **pleasant stay**. We got a large room with 2 **double beds** and 2 **bathrooms**, The TV was Ok, a 27' CRT Flat Screen. The concierge was very friendly when we need. The room was very **cleaned** when we **arrived**, we ordered some pizzas from room **service** and the pizza was Ok also. The **main Hall** is **beautiful**. The breakfast is **charged**, 20 **dollars**, kind **expensive**. The internet **access** (WiFi) is **charged**, 13 **dollars**/day. Pros: Low **rate price**, **huge** rooms, **close** to **attractions** at Loop**, close** to metro station. Cons: **Expensive** breakfast, Internet **access charged**. Tip: When **leaving** the **building**, always use the Michigan Av exit".*

We made one single document by combining spam and non-spam reviews and performed pre-processing steps (tokenization and stop-words removal). In next step, we selected noun, adjective, verb, adverb and interjection for example husband (noun), stayed (verb), very (adverb) and soft (adjective) etc. are select. In next step, we performed the WSD using Roget Thesaurus and generated all sense of word and relation of words based on the ELKB described two relation parameter. In last step, Set and store all noun, verb, adverb, adjective and interjection as candidate word (CW) and filtered other words. In present spam and non-spam example below, bold words are candidate words. The candidate words selection is based on sense of word with relation to words. For example: In spam review example the word "stay" appeared two times with same sense of meaning. The "stayed" word used in first line of example means "To remain at same place". In third line the word "stay" means "To remain at same place". So, this word contains the "same sense" and "repetition" relation of word. The construction of lexical chain is based on the set of candidate words (CWj) where $1 \leq j \leq n$. If any candidate word hold the ELKB Thesaurus two set relation then store that candidate word into respective lexical chain (already created chain) otherwise create a new lexical chain. Set of all lexical chain is called the global set (G). There are total 17 lexical chains generated showed in Table-I.

The notable point here is that, occurrences of each word in lexical chain is depending upon the two relations used for the implementation of lexical chains using the ELKB. The relations are:

1. Repetition of the same word, for example: *stay, stay*.

2. Inclusion in the same Paragraph.

	Table-I Generated Lexical Chains							
L1	stayed	stay	stay	arrived	leaving	stay	staying	staying
L2	double	Hall	dollars	touch	loved	loved		
L3	march	charged	charged	close	close	charged		
L4	friendly	friends	loved	loved				
L5	beds	bathrooms	bedrooms	bathroom				
L6	appreciated	loved	loved	recommend				
L7	main	huge	absolutely					
L8	service	enjoyed	service					
L9	Hall	free						
L10	building	hotel						
L11	access	access						
L12	expensive	price						
L13	beautiful	beautiful						
L14	pleasant	attractions						
L15	nights	nights						
L16	view	minute						
L17	rate	price						

In Table-I, each repeated words is occurring in single chain (according to above defined 1 relation). For example, look at the first lexical chain L1, here we have seen that, all words related to " stay " (lemmatization is not performed) are presented only in single chain. If you saw the, lexical chain L2, L4, and L6, the occurring word " loved" is present in three chains. The reason is that, in L2 chain the word " touch " is a inclusion relation with word "loved" in the same paragraph. Same way other words like, "friends" in L4 chain and "appreciated", "recommend" in lexical chain L6 is having same above relation.

Table-II Lexical Cohesion of Lexical Chains
L1 = Stay→stay (Repetition); arrived →leaving(antonym); staying →staying(Repetition)
L2 = loved→loved(Repetition); double →dollars(collections); hall→ touch(collocations) double → hall(collections); touch →dollars(collocations)
L3 = charged →charged(Repetition); close→close(Repetition); close→march(collocations)
L4 = friendly→friends(Repetition); loved→loved(Repetition)

Table-II shows lexical cohesion of lexical chains (Here we gave only four example of lexical chain for space complexity reason). Lexical chain L1 shows the stay →stay which is followed by "repetition" cohesion property, arrived →leaving is followed by "antonym" and staying→ staying is followed by "repetition". In the same way, other lexical chains (Showed in Table-II) contains cohesion property.

- Example of TF_IRF Features Vector Generation on Spam and Non-Spam Review

Here, we have presented the calculation of first lexical chain L1 with help of an example.

We took first lexical chain from Table-I showed below:

L1= [stay, stay, arrived, leaving, stay, staying, staying]

Here, we first calculated the term-frequency (tf), review-frequency (rf), inverse-review-frequency (irf) and tf-irf of spam and non-spam review example shown below. Here, we have taken every word from lexical chain one by one and counted the frequency (f) of words appeared in spam and non-spam review then we calculated the sum of all counted words. The examples are shown below:

Lexical chain L1 each word counts in spam review example:

[('stayed',1), ('stay', 1), ('stay', 1), ('arrived', 0), ('leaving', 0), ('stay', 1), ('staying', 2), ('staying', 2)]

total number of count=8

tf-spam-review= log\ [1+count]

tf-spam-review=0.954

rf(no. of reviews in which lexical chain appears)=2

irf = 1/log(1+ rf(L))

irf = 2.096

tf-irf= 2

Lexical chain L1 each word counts in non-spam review example:

[('stayed',0), ('stay', 2), ('stay', 2), ('arrived', 1), \\ ('leaving', 1), ('stay', 2), ('staying', 0), ('staying', 0)]

total number of count=8

tf-spam-review= log\ [1+count]

tf-spam-review=0.954

rf(no. of reviews in which lexical chain appears)=2

irf = 1/log(1+ rf(L))

irf = 2.096

tf-irf= 2

When we applied the algorithm-3 on all 17 lexical chain (showed in Table-I) then the final features vector is show in Table-III. This feature vector showed in Table-III used as an input to supervised algorithms classifier.

A. Bag-of-Word Model

The Bag-of –Word (BOW) is very common model which is mainly used for a document is represented as the bag (many sets) of its words. It is mostly used for the purpose of document classification. Where, the frequency of appearing of each word is used as features for training a classifier. In this paper, we have applied the BOW model on 800 reviews and calculated the features vector by TF_IDF algorithm and same is calculated for lexical chain model. The main difference between lexical chain and BOW model is that lexical chain is made on the basis of semantic relation of words occurring co-together. Lexical chain holds the semantic meaning of the words whereas BOW does not hold this relation. The BOW model is built on the basis of selection of raw words. These raw words do not provide any useful information related to words which are bounded with semantic meaning. In results section, we have given comparison of lexical chain model and BOW model output on 800 negative reviews dataset.

Table-III Features vector on spam and Non-Spam Example

Lexical-Chain	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Spam-Review	2	1.6	0	1.6	1	2.8	0.6	1.2	0.6	0.6	0	0	1	0	1	0.6	0
Non-Spam Review	2	1.4	3.9	0.6	1	0	1	1	0.6	0.6	2.3	1.5	1	1.5	1	0.6	1.5

IV. EXPERIMENT SETUP AND RESULTS DISCUSSION

The 400 authentic and 400 manipulative hotel negative reviews (labeled review in spam or non-spam) has used for experiment purpose. We have used open source tool "Rogets Thesaurus based Electronic Lexical Knowledge Base (ELKB)" for lexical chain generation. The BOW model is performed simple as in python 2.7 [23].

The Feature Vector (FV) matrix of lexical chain is 800×216 created though algorithm-3. This matrix is used as an input to train the classifier. This feature vector matrix of lexical chain has already given as the input in WEKA tool [24] for the purpose to run the classification algorithms. We have used Addboost, Bayesian logistic-regression, Function Tree,J48, Jrip, Logistic, Navie Bayes, PART, Random Forest and SVM supervised algorithms as a classifier to train and test the proposed model. The same procedure is used for the BOW model and trained the classifier on above showed algorithms.

All ten supervised methods have run using 10-fold cross validation. The 10-fold cross validation divides the dataset into 10 sections (10 folds) and calculation runs 10 times on dataset. Every time the calculation is run, it prepares the training on 90% of the data and 10% of the test, and every iteration of the calculation, is to change the 10% of the data for training and testing.

The confusion matrix values have defined in such way like, "review is true positive (TP)" when the fake review is classified correctly, "false negative (FP)" when fake review is incorrectly classified and same way "true negative (TN)" when non-fake review correctly classified, otherwise "false positive (FP)" when non-fake incorrectly classified.

Precision is calculated as:

$$\text{Precision}= TP/(TP+FP) \qquad (1)$$

Recall is calculated as:

$$\text{Recall} = TP/(TP+FN) \qquad (2)$$

F-measure is combined Precision and recall.

$$F1 = 2\times(\text{Precision}*\text{Recall})/(\text{Recall} + \text{Precision}) \quad (3)$$

Where the Accuracy is calculated as:

$$\text{Accuracy}= (TP+TN)/ (TP+FP+FN+TN) \qquad (4)$$

We have compared the F-measure and accuracy results of LCBSS algorithm and BOW model, which is shown in Fig.1 and Fig.2. In Fig.1 and Fig.2, LCBSS gives the highest value on all ten supervised algorithms other than BOW model. The SVM gave the 99.75% accuracy for LCBSS algorithm.

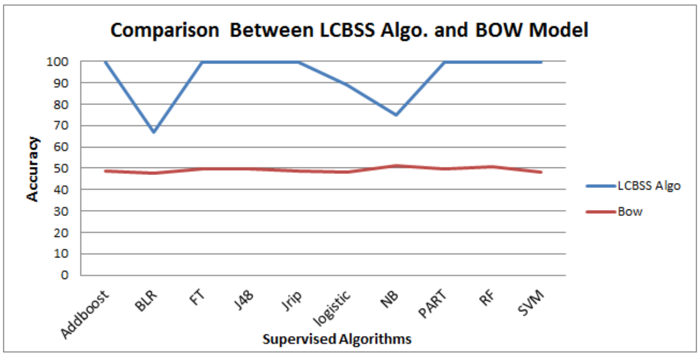

Fig.1: Accuracy Comparison Between LCBSS Algorithm and BOW Model

This shows proposed model accurately classified reviews as a spam or non-spam with greater margin. The receiver operating characteristic (ROC) is a graphical plot that delineates the execution of a binary classifier framework as its separation limit is differed. The curve is made by plotting the true positive rate (TPR) against the false positive rate (FPR) at different limit settings. We have represented the ROC curve in Fig.4. In this figure, the algorithms are represented by the alphabets A,B,C,D,E,F,G,H. Here, D,F represents the perfect classification with highest value. The D and F algorithms gave low false positive rate and high true positive rate which is represented in the algorithms which is perfectly classified in spam or non-spam reviews. The lexical chain model shows better classification on all supervised algorithms. The true positive rate for all algorithms is near to one which represents the lexical chain model accurately classified review as a spam or non-spam.

In Table-IV, we have presented the Support Vector Machine (SVM) Technique Performance Comparison Between Proposed Lexical Chain Model, Used BOW Model and Baseline [16]. The Table-IV Shows the Precision, Recall, F-Measure and Accuracy of all three models. Here, the proposed lexical chain model has given 99.75% highest accuracy among all three models.

Table-IV. SVM Technique Performance Comparison between Proposed LCBSS Algorithm, Used BOW Model and Baseline [1]

Approach	Precision	Recall	F-Measure	Accuracy
Proposed LCBSS Algorithm	**0.998**	**0.998**	**0.998**	**99.75%**
Used BOW Model	0.472	0.333	0.391	48.06%
Baseline[1]	0.864	0.855	0.859	86.00%

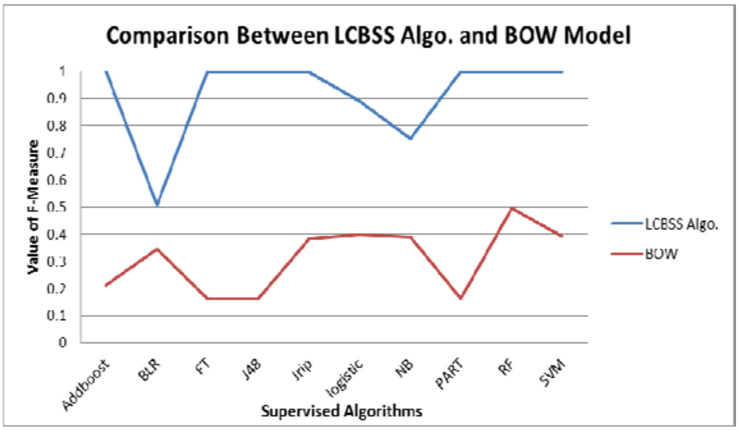

Fig.2. F-Measure Comparison Between LCBSS Algorithm and BOW Model

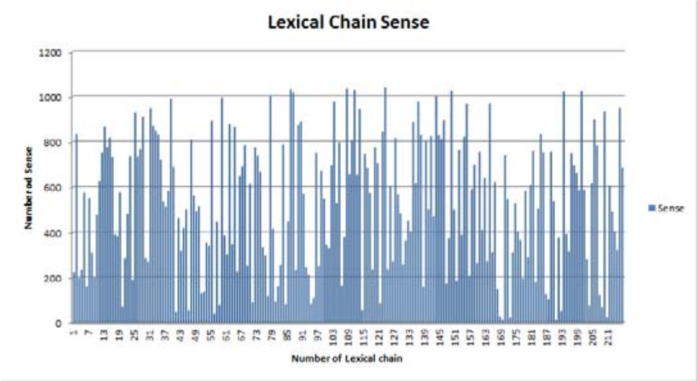

Fig.3 Number of Sense of Lexical Chain

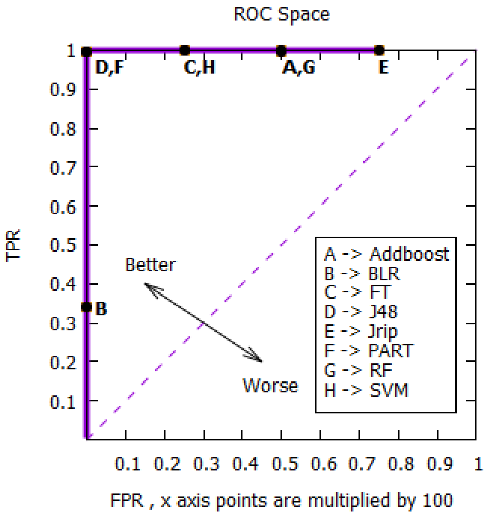

Fig.4 ROC Curve of Supervised Algorithms

V. Conclusion

For on-line booking of hotels, the written feedbacks of other users play the vital role in decision making. If spammer write spam negative reviews, it will be curse for e-commerce industry. In this paper, we have proposed Lexical Chain Based Semantic Similarity (LCBSS) algorithm for negative review spam detection. LCBSS algorithm has generated the feature vector which is used as an input to ten supervised algorithm. In ten supervised algorithm, the SVM has given 99.75% accuracy. The achieved accuracy is highest in between existing techniques. In future work, we can use the proposed model on another dataset. Other knowledge based approaches can also be updated on proposed model.

REFERENCES

[1] Ott, Myle, Claire Cardie, and Jeffrey T. Hancock. "Negative Deceptive Opinion Spam." In HLT-NAACL, pp. 497-501. 2013.

[2] Jindal, Nitin, and Bing Liu. "Opinion spam and analysis." In Proceedings of the 2008 International Conference on Web Search and Data Mining, pp. 219-230. ACM, 2008.

[3] E. Allen. Dear staff. we could do with some positive comments: Hotel boss is caught telling his workers to post fake reviews on tripadvisor. MailOnline (2012, October 9). Internet: http://www.dailymail.co.uk/new s/article-2214974/Hotel-boss-caught-telling-workerspost-fake-reviews-TripAdvisor. html [Jan, 2016], 2012.

[4] Banerjee, Snehasish, and Alton YK Chua. "Applauses in hotel reviews: Genuine or deceptive?." In Science and Information Conference (SAI), 2014, pp. 938-942. IEEE, 2014.https://en.wikipedia.org/wiki/Flesch%E2%80%93Kincaid_readability_tests#Flesch.E2.80.93Kincaid_Grade_Level access date: May2015.

[5] Banerjee, Snehasish, and A. Y. Chua. "A linguistic framework to distinguish between genuine and deceptive online reviews." In Proceedings of the International Conference on ICWS. 2014.

[6] Hu, Nan, Indranil Bose, Noi Sian Koh, and Ling Liu. "Manipulation of online reviews: An analysis of ratings, readability, and sentiments." Decision Support Systems 52, no. 3 (2012): 674-684.

[7] Zheng, Rong, Jiexun Li, Hsinchun Chen, and Zan Huang. "A framework for authorship identification of online messages: Writing-style features and classification techniques." Journal of the American Society for Information Science and Technology 57, no. 3 (2006):378-393.

[8] Banerjee, Snehasish, and Alton YK Chua. "A study of manipulative and authentic negative reviews." In Proceedings of the 8th International Conference on Ubiquitous Information Management and Communication, p. 76. ACM, 2014.

[9] Baumeister, Roy F., Ellen Bratslavsky, Catrin Finkenauer, and Kathleen D. Vohs. "Bad is stronger than good." Review of general psychology 5, no. 4 (2001): 323.

[10] Yin, Dezhi, Sabyasachi Mitra, and Han Zhang. "Research Note—When Do Consumers Value Positive vs. Negative Reviews? An Empirical Investigation of Confirmation Bias in Online Word of Mouth." Information Systems Research 27, no. 1 (2016): 131-144.

[11] Morris, Jane, and Graeme Hirst. "Lexical cohesion computed by thesaural relations as an indicator of the structure of text." Computational linguistics 17, no. 1 (1991): 21-48.

[12] Algur, Siddu P., Amit P. Patil, P. S. Hiremath, and S. Shivashankar. "Conceptual level similarity measure based review spam detection." In Signal and Image Processing (ICSIP), 2010 International Conference on, pp. 416-423. IEEE, 2010.

[13] Jindal, Nitin, Bing Liu, and Ee-Peng 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] Lau, Raymond YK, S. Y. Liao, Ron Chi Wai Kwok, Kaiquan Xu, Yunqing Xia, and Yuefeng Li. "Text mining and probabilistic language modeling for online review spam detecting." ACM Transactions on Management Information Systems 2, no. 4 (2011): 1-30.

[15] Sandulescu, Vlad, and Martin Ester. "Detecting Singleton Review Spammers Using Semantic Similarity." In Proceedings of the 24th International Conference on World Wide Web, pp. 971-976. ACM, 2015.

[16] Mukherjee, Arjun, Bing Liu, and Natalie Glance. "Spotting fake reviewer groups in consumer reviews." In Proceedings of the 21st international conference on World Wide Web, pp. 191-200. ACM, 2012.

[17] Ott, Myle, Yejin Choi, Claire Cardie, and Jeffrey T. Hancock. "Finding deceptive opinion spam by any stretch of the imagination." In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1, pp. 309-319. Association for Computational Linguistics, 2011.

[18] Jarmasz, Mario. "Roget's thesaurus as a lexical resource for natural language processing." arXiv preprint arXiv:1204.0140 (2012).

[19] Silber, H. Gregory, and Kathleen F. McCoy. "Efficient text summarization using lexical chains." In Proceedings of the 5th international conference on Intelligent user interfaces, pp. 252-255. ACM, 2000.

[20] Silber, H. Gregory, and Kathleen F. McCoy. "Efficiently computed lexical chains as an intermediate representation for automatic text summarization." Computational Linguistics 28, no. 4 (2002): 487-496.

[21] Halliday, Mn AK. "&R. Hasan. Cohesion in English." (1976).

[22] Jayarajan, Dinakar, Dipti Deodhare, and Balaraman Ravindran. "Lexical chains as document features." (2008): 111-117.

[23] (2016) The Python 2.7.0 release [Online]. Available: https://www.python.org/download/releases/2.7.

[24] Hall, Mark, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. "The WEKA data mining software: an update." *ACM SIGKDD explorations newsletter* 11, no. 1 (2009): 10-18.

AUTHOR PROFILE

Rupesh Kumar Dewang is an assistant Professor at Computer Science and Engineering Department of Motilal Nehru Institute of Technology Allahabad, India. He has completed her Bachelor’s and Master’s degree from RGPV Bhopal India. He has a teaching experience of more than 07 years. He has published 04 papers in international conferences and 01 paper in international journal. His area of research is data mining, sentiment mining and big data & cloud computing security. Several bachelors and masters students are currently working under his supervision.

Dr. Anil Kumar Singh is an associate professor at Computer Science and Engineering Department of Motilal Nehru Institute of Technology Allahabad, India. He has completed his Ph.D. from IIT Roorkee. He has a teaching experience of more than 20 years. His area of interest includes Semantic analysis, Big data and Cloud computing. He has published papers in different national, international conferences and journals. Several masters and Ph.D. scholars are currently working under his supervision