

# Opinion Score Mining: An Algorithmic Approach

**Surbhi Bhatia<sup>1</sup>**

<sup>1</sup>Banasthali University/ CS Department, Rajasthan, India  
E-mail: surbhibhatia1988@yahoo.com

**Manisha Sharma<sup>2</sup> and Komal Kumar Bhatia<sup>3</sup>**

<sup>2</sup>Banasthali University/CS Department, Rajasthan, India  
<sup>3</sup>YMCA University of Science and Technology/ CSE Department, Haryana, India  
E-mail: manishasharma8@gmail.com<sup>2</sup>, komal\_bhatia1@rediffmail.com<sup>3</sup>

Received: 03 April 2017; Accepted: 07 July 2017; Published: 08 November 2017

**Abstract**—Opinions are used to express views and reviews are used to provide information about how a product is perceived. People contributions lie in posting text messages in the form their opinions and emotions which may be based on different topics such as movie, book, product, and politics and so on. The reviews available online can be available in thousands, so making the right decision to select a product becomes a very tedious task. Several research works has been proposed in the past but they were limited to certain issues discussed in this paper. The reviews are collected which periodically updates itself using crawler discussed in our previous work. Further after applying certain pre-processing tasks in order to filter reviews and remove unwanted tokens, the sentiments are classified according to the novel unsupervised algorithm proposed. Our algorithm does not require annotated training data and is adequate to sufficiently classify the raw text into each domain and it is applicable enough to categorize complex cases of reviews as well. Therefore, we propose a novel unsupervised algorithm for categorizing sentiments into positive, negative and neutral category. The accuracy of the designed algorithm is evaluated using the standard datasets like IRIS, MTCARS, and HAR.

**Index Terms**—Opinion, Mining, Crawler, Unsupervised Learning, Sentiment Analysis.

## I. INTRODUCTION

With the growth of e-commerce and due to the explosion of blogs and social media, web has become the richest source of online shopping. The popularity of these sites has fascinated the people to such an extent that 76% of the users are actively participating in the social mobility around the globe [1]. The attractive offers induced by the social networking sites have fascinated customers to buy the products online. But with the thousands and hundreds of online reviews available online, to make a right decision regarding buying of a product has become a tedious task to achieve. Reading few reviews will lead to biased reviews and finally a wrong decision. The possible solution to this problem is

using recommender systems [2]. But, these also solely rely on making the user believe the underlying algorithm blindly failing to help the people to identify the useful information effectively [3].

Early work in Review mining focused mainly at the document and the sentence level which discard some of the important feature in categorization the reviews into positive and negative category. By, to classify the documents at the finer level i.e. the aspect level where all the features in the review are identified and the overall goodness or badness score is calculated using certain functions is lacking [4]. Over the last few years, Opinion mining has gained its importance in terms of research. But it has always been proved that human proposed methods are less efficient than machine learning approaches [5]. Many researchers used supervised learning methods like Naïve Bayes, Maximum entropy, and Support Vector Machines for classifying sentiments on different domains.

Traditionally, text classification was mainly done by humans, features developed by them, consulting dictionaries, knowledge based techniques or customized hierarchical component like tree kernels. These methods have not proved to be much efficient in today's fast developing phase. So, in contrast to traditional approaches, we present a novel unsupervised learning approach for classifying sentiments into positive, negative and neutral category. The algorithm's efficiency is compared with the famous K means algorithm. The accuracy of the proposed technique using various datasets are compared and illustrated in graphs and charts. The paper has been organized as follows:

The remainder of the paper is organized as follows: Section 2 discusses the related work. Section 3 explains the work proposed with the framework and its algorithms. In section 4, the experimental evaluation is conducted and the analysis is thoroughly discussed and Section 5 concludes the proposed work.

## II. RELATED WORK

Various researchers have focused their research on determining sentiments from reviews and make them fall

in one of the three categories, i.e. positive, negative and neutral. Our approach is to classify the reviews using simple, easy to adapt and applicable technique as discussed in the next section. Some of the related works done by the prominent researchers and the issues in this field are discussed below:

Shu Zang [6] focused on identifying the product features based on supervised learning technique by combining the frequency of the words used, syntax tokens and domain knowledge. The author used the technique of data mining for finding frequent features, i.e. association rule mining. Linguistic rules were applied to determine semantic orientation. The advantages were that the work proposed identified not only the opinionated features but also the positioning correctly in the positive, negative and neutral category with improved performance. The disadvantages were that the low frequency features tend to be missing, opinion words were not in the close vicinity with the features. The approach can be improvised on the following two techniques. Syntactic analysis or pattern analysis method can be taken for finding subsequent opinion words. Also, the statistic information can be combined with polarity calculation method.

Raisa Varghese [7] explored the aspect based opinion mining by combining the use of dependency parsing, conference resolution and senti word net. Support vector machine classifier is used for the training of the data on a single domain. The results were promising and the novelty of the work which uses the above combinations for relation extractions proved to be efficient. The limitations of the above work are summarized as: Sentences with sarcasm are not considered, pre-processing tasks of the well-formed natural language structure needs to be defined.

Li Zhang [8] proposed a novel method based on slack function, sentiment dictionary and mini distance for identifying aspects and assigning polarities. The advantages are that slack approach reduces the scope of potential evaluation objects. Evaluation objects are extracted by using POS tagger to detect NP nodes which is done at the sentence level. Further improvement can be followed up by using association method to predict the emotional preference

A.Jeyapriya [4] proposed a novel system in order to identify aspects using frequent item set mining for categorizing the online reviews into positive and negative category. The architecture developed performs a finer grained analysis using naïve based classification algorithm by using opinion word rule. The advantages are that the overall sentiment orientation using supervised term counting based approach gives a good accuracy. The work proposed can be extended by summarizing the aspects based on the relative importance of the extracted aspect.

Deepak Kumar Gupta [9] proposed an ensemble system using different classifiers and combined them by using a majority voting technique. Post processing is performed on each classifier with a set of heuristics. Random forest classifier was used as a classification model. The overall

system worked well for the positive and negative classes but suffers most for the conflict class. The paper has certain limitations because of the misclassifications of aspect terms to others.

Shamita Pisal [10] proposed a search engine that focuses on detecting and mining the opinion words that determines the polarity information. The overall summary is presented from the features in the sentiments reviewed by the opinion ranking algorithms. The review score is calculated by the proposed HAC algorithm (Highest count algorithm), which can be tuned appropriately according to the dataset on which the algorithm is applied. Another advantage of this paper is that the reviews with their features are determined in an attractive and spontaneous manner and the ratings are also shown in the summary. The work is limited to only one review site which can be further extended. The prototype can also be expanded.

Farhan Hassan Khan [11] proposed lexicon-based methodology with machine learning for sentiment analysis. The proposed SKBs help in determining the sentiment orientation, subjectivity, and sentiment strength and word sense. The paper well explained Mathematical models for sentiment strength computation using sentiwordnet and compared and analysed the results by taking different datasets.

[12] Muqtar Unnisa proposed automated approach of unsupervised learning as spectral clustering is used approach on Twitter data. Experiments were conducted by taking movie review datasets and the link between the two clustering techniques i.e. spectral and k-means clustering are shown hypothetically. The results have been shown in graphs and comparison has been made to obtain high quality results. The algorithm proposed by the author proved to be scalable but the work was limited to work on tweets and may not give promising results in large datasets.

### III. PROPOSED WORK

The work proposed is explained in various modules.

#### A. Data Retrieval

The relevant opinions are extracted from the dynamically generated result pages on submitting a query. A novel technique for crawling the web pages, which is periodically updated in order to extract opinions by computing the dynamic priority for each site, has been developed that results in more relevant and fresh web pages containing opinions which is discussed in our previous research [13]. In the data retrieval module, design of opinion crawler has been proposed that not only downloads the current opinions of various products of the user but also crawls the opinions incrementally so that to keep the repository fresh with the latest/refresh opinions. The reviews are collected domain wise. The whole process of data retrieval [13] is explained in the architecture explained below in Fig. 1.

URL Database consists of the URLs which are the seed URLs. The URLs are stored in the form of the queue,

with the timestamp information associated with the respective URL. The timestamp contains the average time difference of the existing opinions and the new opinions. The URL with the lowest timestamp has the highest probability of having new opinions. *SeedURLs* are given the timestamp 0. URL Fetcher extracts the *SeedURLs* and sends the *URLAvail* signal to the Opinion Tracker to fetch the URL from the *URLRegister*. Opinion Tracker checks if the URL contains the opinions by checking for the span tag (URL) or div tag (URL) and further checks for the comment, review text, summary or opinion in the HTML code extracted.

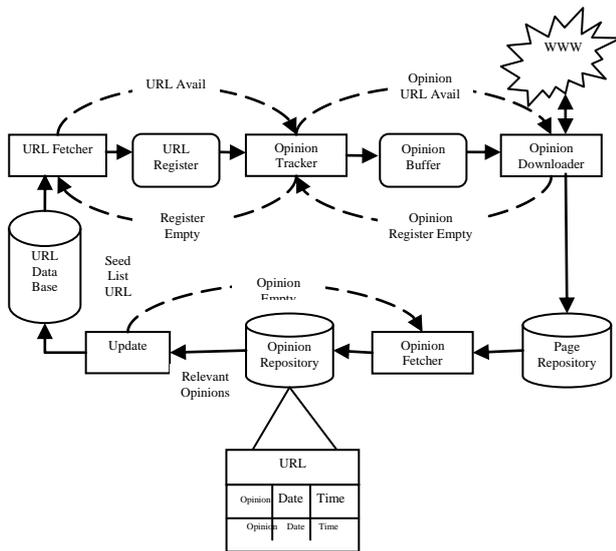


Fig.1. Data Retrieval

If it is a valid URL, Opinion Tracker send the *OpinionURLAvail* signal to the Opinion Downloader. Opinion Downloader extracts the URL stored in the *OpinionBuffer* and downloads the corresponding web pages from World Wide Web and stores it in the Page Repository, otherwise Opinion Downloader sends the *OpinionRegisterEmpty* signal to the Opinion Tracker. Next, the Opinion Fetcher extracts the opinions from the web page by analyzing the structure of the HTML page and using the string matching technique. The extracted opinions are stored in the Opinion Repository with the link information as date and time of when the opinion was posted. The Update module further extracts the URLs which are stored individually in a file. The Update module calculates the average timestamp of each individual URL and stores them in the URL database with the timestamp information associated with the respective URL. In next cycle, the URL Fetcher fetches the URL with the lowest timestamp and the process continues.

For example. On submitting the query “Apple I Phone-4S”. The following steps are undertaken.

- Crawling item from the seed URL “www.Amazon.com*
- Separate Files created for each new link found*
- Crawling Reviews of first link (Page 1)*
- Reviews crawled for all pages*

*Sample reviews collected*  
*Estimated time to revisit the URL.*

The example is shown in the given snapshot below in Fig. 2 and Fig. 3 by taking the URL [www.amazon.com](http://www.amazon.com).

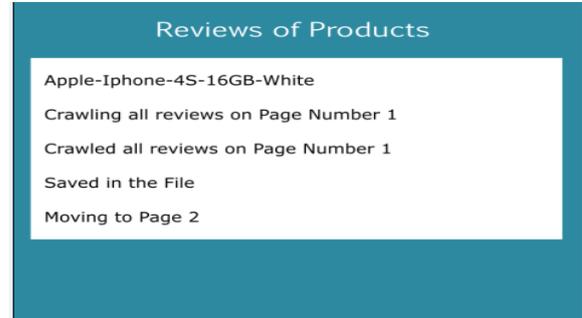


Fig.2. Snapshot of reviews Fetching



Fig.3. Crawling of reviews

**B. Pre-processing**

The reviews which are extracted using opinion crawler are then received as input to Pre-process Opinions module. The pre-processing tasks involve cleaning operations in order to find the relevant content from opinion repository to get the final result. The output will be stored in Filtered Opinion Repository.

*Pre-Processing Tasks*

The following are the pre-processing tasks applied to the opinions and are stored in the filtered opinion repository.

*Tokenization:* It is the task of splitting sentences into pieces, called tokens, perhaps at the same time throwing away certain characters, such as punctuation. Tokenization will make our review sentences to split in tokens. The part-of-speech tagging will categorize the English grammar in nouns, verbs, adjectives, pronouns, prepositions, conjunctions and interjections. Each word (token) will be labeled with its appropriate part of speech. The POS tagging will be helpful later in the determination of features and opinion words from the reviews. It can be done manually or with the help of the POS tagger tool. Our work includes Stanford tagger which tag each word in an online review sentence because manual tagging will be very time consuming. For POS tagging the documents, we used Stanford Natural Language Processing Parser [14], which POS tag the all the documents and generates POS tagged XML doc. as an

output file. In the examples “This is good phone” shown below:

The sentences is tagged as, good is tagged with tag JJ which indicates ‘good’ is an adjective where a ‘phone’ is tagged as NN which indicates noun given in Fig. 4

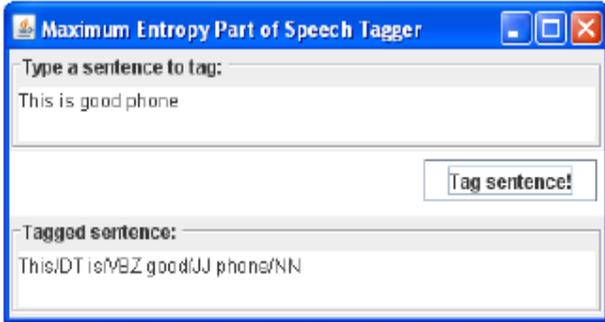


Fig.4. Output of Stanford POS Tagger

*Stop word elimination:* It is a process of eliminating the commonly occurring or rarely occurring words existing in a sentence like a, an, the, are etc. Stop word removal is another important task in data preprocessing. Stop words does not carry any relevant information and are language specific functional words [4]. These are helpful in removing unwanted words in each review sentence which can be checked against stop word list. It can be of types like pronouns, prepositions, conjunctions etc.

*Stemming:* It usually refers to a crude heuristic process that cuts off the ends of words, which may often include the removal of derivational affixes. Stemming ensures to get the root word of the word. This algorithm works on reducing the root word “make” from the words like “making”, “maker” and “makes”. Many algorithms come under stemming like Affix stemmers, lemmatization, n gram analysis, porter stemmer etc.

*Lemmatization:* It usually refers to doing things properly with the use of a vocabulary and morphological analysis of words, normally aiming to remove inflections only and to return the dictionary form of a word, which is known as the lemma. If confronted with the token saw, stemming might return just s, whereas lemmatization would attempt to return either see or saw depending on whether the use of the token was as a verb or a noun.

*Normalization:* It is a process of canonicalizing tokens so that matches occur despite superficial differences in the character sequences of the tokens. For instance, if the tokens anti-discriminatory and antidiscriminatory are both mapped onto the term antidiscriminatory.

The proposed design explains the detailed structure of processing opinions. The steps are explained under the following Fig. 5.

It consists of mainly three steps.

- Opinion Extractor
- Tokenizer
- SSLN (Stemming, Stop Word Removal, Lemmatization, Normalization)

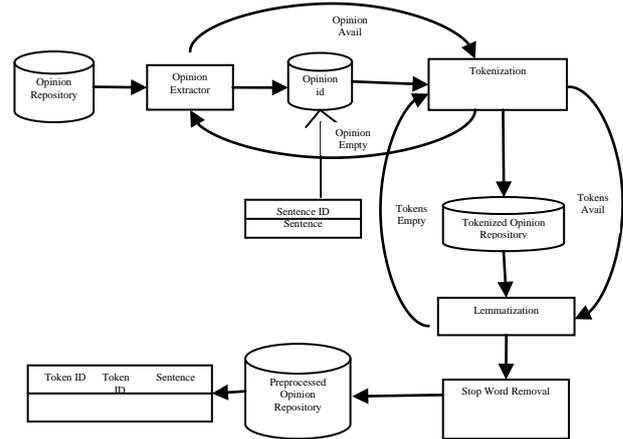


Fig.5. Opinion Extractor

*Opinion Extractor:* This module will extract all the opinions from the opinion repository and will break the opinions into individual sentences with their particular sentence ID. The sentences will be stored in the Opinion ID Repository. This will further send the Opinion Avail signal to the Tokenizer to tokenize the sentences. If no sentences are available in the Opinion ID Repository, then Tokenizer will send Opinion Empty signal to the Opinion Extractor. The algorithm is given in Table 1.

Table 1. Algorithm of Opinion Extractor

```

Opinion Extractor ()
{
wait(Opinion Avail)
fetch Opinions from the Opinion Repository
break Opinions into individual sentences with Sentence ID
Store in Opinion ID Repository
signal (Opinion Empty)
}
    
```

*Tokenizer:* Tokenization will make our review sentences to split in tokens. For POS tagging the documents, we used Stanford NLP Parser [14], which POS tag the all the documents and generates POS tagged XML doc as an output file. The tokens will be stored in the Tokenized Opinion Repository. Tokenizer will send the TokensAvail Signal to next component, i.e. SSLN; else SSLN will send the TokenEmpty signal back to Tokenizer. The algorithm is shown in Table 2.

Table 2. Tokenizer

```

Tokenizer()
{
wait(Tokens Avail)
Fetch Sentences from the Opinion ID Repository
Tokenize the sentences into tokens
Store in Tokenized Opinion Repository
signal(Tokens Empty)
}
    
```

*SSLN:* This constitutes four tasks to achieve. Lemmatization, Stop Word Removal, Stemming and Normalization. All these tasks (shown in Figure 4.5) are performed and tokens with its token id and sentence is stored in Filtered Opinion repository.

Table 3. Algorithm for SSLN

```

SSLN()
{
wait(Tokens Avail)
Fetch Tokens from the Tokenized Opinion Repository
Apply Lemmatization to tokens
Apply stop word removal
Apply stemming
Apply normalization
Store in Preprocessed Opinion Repository
signal(Tokens Empty)
}

```

Finally Filtered Opinion Repository will contain the pre-processed tokens with the token ID and Sentence ID.

### C. Sentiment Classification

We extract only nouns from the pre-processed opinion directory because nouns are considered as the most appropriate target for which the opinion is written [15]. So, opinion word list will be formed by finding the adjective to the closest noun.

Based on this word list, we will categorize the sentences into positive, negative and neutral category. The algorithm we have proposed is based on unsupervised learning technique called as Opinion Score Mining (OCM).

*Algorithm : Opinion score Mining (OCM):*

#### Step 1:

Take two sets of words:

Name it,

Array 1: Positive

Array 2: Negative

Array 3: Neutral

Example:

Positive [n] = [good, better, best]

Negative [n] = [bad, worse, worst]

Neutral [n] = [the, phone, my, is]

Conditional[n] = [better, neither, nor, worse, but]

#### Step 2:

Set the scores for each word. This is done using some user defined rules.

For example: rule could be as follows:

negative[n] = negative scores

positive [n] = positive scores

neutral [n] = zero

#### Conditions:

1. There is no basis for this rule, We simply set a score for the words on the array.

2. The magnitude of negativity or positivity is determined using the unsupervised learning of the algorithm, where the user first manually gives scores to word that gets stored in a DB or Table, which is later used as a reference table for scoring.

#### Step 3:

Iteration 1: Simple sentences

1. Get sentence:

Sentence = the lens works good

2. Store it in an array Sentence[n] :

The lens works good Sentence:

[0] [1] [2] [3]

3. Check for words coming from either of the two defined sets,

If Negative[n] == Sentence[k]

Display: Detected Negative (-ve) Response Set: -ve Score (set as per the table).

Elif Positive[n] == Sentence[k]

Display: Detected Positive (+ve) Response Set:+ve Score (set as per the table).

Elif Neutral[n] == Sentence[k]

Display: Detected Neutral Response. Set: Zero Score (set as per the table).

Else(#)

Display: None detected, press ALT+ENTER for review.

4.(#)If for a sentence "Sentence": Positive-Score = Negative-Score,

Then,

Display: Score error in "Sentence". Press CTRL+8 to review.

#### Critical Scenario One:

Sentence = the lens is pretty bad.

Now, the algorithm fails at when it hits one or both of the two cases (Marked with a '#' above).

Scenario One:

Display: None detected, press ALT+ENTER for review.

This can be because the DB or Table doesn't contain the words or phrases used.

In that case, we set the scores for the words and phrases and the DB or Table gets updated ON-THE-GO [Using online training of algorithms for a non-continuous data stream].

#### Scenario Two:

If for a sentence "Sentence":

Positive-Score = Negative-Score, Then,

Display: Score error in "Sentence". Press CTRL+8 to review.

This is a failure that happens when we HAVE the required words or phrases in the DB or Table, but for some linguistic difficulties, the algorithm hasn't been able to check for the spaminess (Negativity) or the haminess (Positivity) of the sentence.

In this case we make another database or table and we call it a confusion matrix. This matrix will contain all the words or phrases which possibly created confusion in the past and based on what it learnt from the past, a prior probability model is generated from which we see the maximum likelihood of any word or phrase.

In this way, we do not contaminate the original reference table but we keep track of all the non-standard words or phrases and use them whenever the conventional algorithm fails.

### Scenario Three:

A comparison scenario:

Sentence = the lens A is better than lens B.

Or

Sentence = Lens A is good, but B is better.

In this case, the algorithm will be unable to correctly get a hit for a positive for a negative phrase.

If, such a case occurs, the algorithm shall pop the sentence for review.

Since, the OCM is Unsupervised, the only way to handle this is to either always keeps a reviewer for critically formed sentences, but our approach is not giving any training data so we make a rule for this kind of sentences to be handled.

One possible suggestion for such a rule is as follows:

1. Tokenize “Lens A” and “Lens B”, then if the token of “Lens A” comes after “Lens B” [example: Lens B is better than Lens A], then “Lens B” gets a positive score.

2. If the token of “Lens A” has a negative word associativity [example: Lens A is not better than Lens B], then “Lens B” gets a positive score.

3. If both the tokens have negative word associativity [example: Neither Lens A, nor Lens B is good], then both of them get 0 score.

Few cases have been explained in detail. OCM is flexible enough to incorporate other cases as well. This novel unsupervised algorithm, OCM will categorize and assign the score to each sentence.

The scores assigned to each word (token) are done as follows:

Checking each token with the dictionary meaning and Senti Wordnet make our method of assigning scores unique and more efficient. Senti Wordnet is particularly developed for opinion mining applications and is an outstanding freely available lexicon available online. The polarity scores are shown with the words in Princeton WordNet in technical terms [16]. They particularly assign polarity scores to each Sysnet in the Wordnet [17]. Use of dictionary will help in finding synonyms and antonyms relations of the words which are not present in the Opinion lexicon for determining the polarity of new opinion words. These words will be added to our database.

## IV. RESULTS AND ANALYSIS

Next is to perform analysis of our algorithm. IRIS plant dataset is considered as one of the most suited for the problem of classification. It has been found from UCI Machine Learning Repository and created by R.A. Fisher. There are three classes of about 50 objects or data values each class signifying the category of IRIS plant [18]. So for instance, altogether 150 objects are collectively taken which is categorized similarly in three classes. The three classes are named as: IRIS Setosa, IRIS Versicolour, or IRIS Virginica [19]. There are four numeric attributes; all are of the continuous form given below: sepal length,

sepal width, petal length, petal width.

### A. Advantages of our algorithm over K-Means:

1. The algorithm designed by us for sorting the tweets can be used as both a classification technique and as a clustering technique, whereas, K-means can only be used as a clustering technique.

2. K-means fails to successfully sort classes when factors/tokens overlap. An example of this case is, “The phone is pretty bad.” In this case “pretty” and “bad” would have different mean points of clustering, this will generate false results. However, in our algorithm, we have designed an unsupervised approach towards this and involve the user’s monitoring to handle such words and update the clusters/classes on a dynamic basis.

We ran our algorithm on Python using a given set of Bad and Good words, the comparison of results as shown in Table 4.

Table 4. Comparison

No. Of Words (MC)	1061	NA
No. Of Words (P)	1077	Counts the empty spaces, this doesn't cause problem in computation because blank spaces are considered as “Useless” words.
No. Of Good Words (MC)	26	NA
No. Of Good Words (P)	24	An accuracy of 92.30769%
No. Of Bad Words (MC)	26	NA

\*MC=Manual Count

\*P= Programmatically

Total score = No. of good words + No. of bad words = 24 + (-23) = 1

### B. Analysis of K-means:

Because K-means is not known as a very robust algorithm for handling text mining problems, we will be using the IRIS dataset for the analysis. The same results can also be reproduced using other standard datasets like HAR, MTCARS or BOSTON-HOUSING and real dataset as ‘Iphone 4S’ on Electronics domain. The query ‘Iphone 4S’ product consists of 980 reviews from Amazon.com with 230 different features. Clearly, the accuracy of k-means is way worse than what our algorithm was predicting, that is clearly because of the unsupervised approach of the designed algorithm.

The proposed approach is evaluated and accuracy, Precision, Recall and F-measure are selected as performance evaluation measures [20].

*Precision:* It is defined as the ratio of the correctly classified opinion words over all the opinion words found. Mathematically, the Precision is given by:

$$P = \text{OWP} \div (\text{OWP} + \text{WOWP}) \quad (1)$$

where OWP is Relevant Opinion words classified and WOWP is the number of Irrelevant Opinion words

classified. It is usually expressed as a percentage.

*Recall:* It is defined as the ratio of the fraction of correctly classified opinion words by the proposed algorithm over all the opinion words. Mathematically, the Recall is given by:

$$P = \text{OWP} \div (\text{OWP} + \text{NOWP}) \quad (2)$$

where OWP Relevant Opinion words classified and NOP Relevant Opinions not classified. It is usually expressed as a percentage.

*F-Measure:* Combining recall and precision values gives F-measure.

$$F = 2PR \div (P+R) \quad (3)$$

where Precision P and Recall R are equally weighted. The achieved accuracy is shown in Table 5.

Table 5. Analysis

Algorithm	Average accuracy
Our Algorithm	96%
K-Means	24%

The analysis has been done using K-Means and proposed approach (unsupervised learning algorithm) in Python and the comparison of the same using real dataset Iphone and standard datasets such as IRIS, HAR, MTCARS is illustrated in Fig. 6.

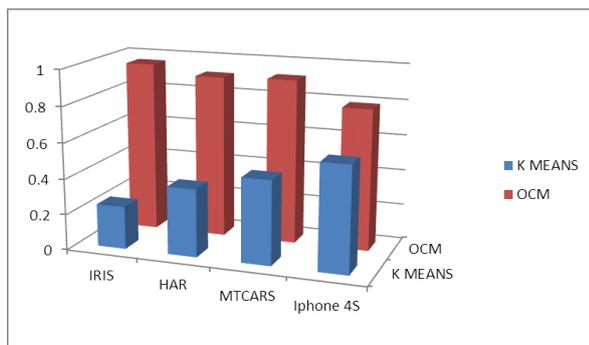


Fig.6. Analysis using datasets

## V. CONCLUSIONS

The parameters taken in behavioral analysis of the social network will raise the accuracy of classifying the opinions into positive, negatives category. The novel unsupervised learning technique is proposed that can solve the problem of domain dependency and reduce the need of annotated training data. The prime objective of our research is to overcome the problem of clustering multiple files with unlabeled data and perform sentiment classification. Our algorithm can be easily applicable to other complex sentences and can be applied to several other domains with sufficiently large corpuses. The major challenge in the field of sentiment analysis is summarization of opinions based on the features

extracted from these opinions. Since features may be represented in different forms of sentiments, so to extract them presents a major research which is still a bottleneck. Rana [21] explained the methods of aspect extraction both explicit and implicit. The various approaches using supervised, unsupervised and semi supervised learning have been discussed and comparison has been made analytically. In future we will propose a complete dynamic system of summarizing these opinions based on the aspects considering both explicit and implicit part of these aspects. We will focus on comparing our algorithm with other existing supervised learning approaches and performance will be evaluated on different datasets.

## REFERENCES

- [1] D. WinderThe importance of social mobilityGemalto Review Magazine, 2010, p. 9.
- [2] Winder D. The importance of social mobility. Gemalto Review Magazine. 2010;9.
- [3] Jotheeswaran J, Loganathan R, Madhu Sudhanan B. Feature reduction using principal component analysis for opinion mining. International Journal of Computer Science and Telecommunications. 2012 May;3(5):118-21.
- [4] Jeyapriya A, Selvi CK. Extracting aspects and mining opinions in product reviews using supervised learning algorithm. InElectronics and Communication Systems (ICECS), 2015 2nd International Conference on 2015 Feb 26 (pp. 548-552). IEEE.
- [5] Pang B, Lee L. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. InProceedings of the 42nd annual meeting on Association for Computational Linguistics 2004 Jul 21 (p. 271). Association for Computational Linguistics.
- [6] Zhang S, Jia WJ, Xia YJ, Meng Y, Yu H. Opinion analysis of product reviews. InFuzzy Systems and Knowledge Discovery, 2009. FSKD'09. Sixth International Conference on 2009 Aug 14 (Vol. 2, pp. 591-595). IEEE.
- [7] Varghese, R., & Jayasree, M. Aspect based Sentiment Analysis using support vector machine classifier. In Advances in Computing, Communications and Informatics (ICACCI), 2013 International Conference on (pp. 1581-1586). IEEE.
- [8] Zhang, L., Xu, W., & Li, S. (2012, September). Aspect identification and sentiment analysis based on NLP. In 2012 3rd IEEE International Conference on Network Infrastructure and Digital Content (pp. 660-664).
- [9] Gupta DK, Ekbal A. IITP: supervised machine learning for aspect based sentiment analysis. SemEval 2014. 2014 Aug 23:319.
- [10] Pisal S, Singh J, Eirinaki M. AskUs: An Opinion Search Engine. InData Mining Workshops (ICDMW), 2011 IEEE 11th International Conference on 2011 Dec 11 (pp. 1243-1246). IEEE.
- [11] Khan FH, Qamar U, Bashir S. A semi-supervised approach to sentiment analysis using revised sentiment strength based on SentiWordNet. Knowledge and Information Systems. 2016:1-22.
- [12] Unnisa M, Ameen A, Raziuddin S. Opinion Mining on Twitter Data using Unsupervised Learning Technique. International Journal of Computer Applications. 2016 Jan 1;148(12).
- [13] Bhatia S, Sharma M, Bhatia KK. A Novel Approach for Crawling the Opinions from World Wide Web. International Journal of Information Retrieval Research

- (IJRR). 2016 Apr 1;6(2):1-23.
- [14] Collobert R, Weston J, Bottou L, Karlen M, Kavukcuoglu K, Kuksa P. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*. 2011, (Aug):2493-537.
- [15] Bafna K, Toshniwal D. Feature based summarization of customers' reviews of online products. *Procedia Computer Science*. 2013 Jan 1;22:142-51.
- [16] Baccianella S, Esuli A, Sebastiani F. SentiWordNet 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining. In *LREC 2010 May 17 (Vol. 10, pp. 2200-2204)*.
- [17] Marrese-Taylor E, Velásquez JD, Bravo-Marquez F. A novel deterministic approach for aspect-based opinion mining in tourism products reviews. *Expert Systems with Applications*. 2014 Dec 1;41(17):7764-75.
- [18] Kolhe SR, Ranjana SZ. Clustering Iris Data using Supervised and Unsupervised Learning. *International Journal of Computer Science and Application*.(2010):0974-767.
- [19] Swain M, Dash SK, Dash S, Mohapatra A. An approach for iris plant classification using neural network. *International Journal on Soft Computing*. 2012 Feb 1;3(1):79.
- [20] Järvelin K, Kekäläinen J. IR evaluation methods for retrieving highly relevant documents. In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval 2000 Jul 1 (pp. 41-48)*. ACM.
- [21] Rana TA, Cheah YN. Aspect extraction in sentiment analysis: comparative analysis and survey. *Artificial Intelligence Review*. 2016 Dec 1;46(4):459-83.

Institute of Engineering, Faridabad. His areas of interests are Search Engines, Crawlers and Hidden Web.

**How to cite this paper:** Surbhi Bhatia, Manisha Sharma, Komal Kumar Bhatia, "Opinion Score Mining: An Algorithmic Approach", *International Journal of Intelligent Systems and Applications(IJISA)*, Vol.9, No.11, pp.34-41, 2017. DOI: 10.5815/ijisa.2017.11.05

## Authors' Profiles



**Surbhi Bhatia** is currently teaching in K.R. Mangalam University, Gurgaon as an Assistant Professor and has almost 7 years of teaching experience. She is pursuing Ph.D in Computer Science from Banasthali University, Rajasthan. She has done M.Tech. in Computer Science (2012) from Amity University and B.E. in Information Technology (2010) from Maharishi Dayanand University. She has published 16 research papers in various International/National Journals. Her areas of interests include Data mining, Sentiment Analysis, Information Retrieval and Genetic algorithms.

**Manisha Sharma** is former Associate Professor of Banasthali University, Rajasthan. She has around 18 years of experience. She has done her Ph.D. in Computer Science from Banasthali University and has published more than 30 research papers in reputed national and international journals. Her main research work focuses on Artificial Intelligence, Data mining and Information Retrieval

**Komal Kumar Bhatia** received the B.E. and M.Tech. degrees in Computer Science Engineering with Hons. from Maharishi Dayanand University in 2001 and 2004 and Ph. D. in Computer Engineering from YMCA university of Science and Technology. Presently, he is working as Professor and Chairman in Computer science and Engineering department in YMCA