International Journal of Multimedia and Ubiquitous Engineering
Vol. 12, No. 7 (2017), pp. 19-18
http//dx.doi.org/10.14257/ijmue.2017.12.7.02

Design and Implementation of Algorithm to Analyze Overall Effect of Customers Reviews

Sartaj Ahmad[1], Ashutosh Gupta[2], Neeraj Kumar Gupta[1], and Rishabh Shukla[1]

[1]*KIET Group of Institutions, Ghaziabad Affiliated to Dr.A.P.J. Abdul Kalam Technical University, Lucknow, U.P., India*
[2]*School of Sciences, U.P.Rajarshi Tandon Open University, Allahabad (UP), India*
sartajahmad2u@gmail.com, neeraj.gupta@gmail.com, rishabh.1313088@kiet.edu, ashutosh3333@gmail.com

Abstract

At present trend of online shopping is increasing very fast in second largest population country like India. Almost shopping sites ask for reviews about the sold products or services provided. Therefore, number of reviews is increasing very rapidly. These reviews are written in free style. Therefore, problem is how to extract knowledge from such type of unstructured reviews. It will be time taken if it is done manually. To overcome such problems two detailed algorithms are proposed in this paper which extracts opinions of customers. This summary of opinions helps new customer before buying a particular product. This paper is divided into four sections one for introduction, second for proposed algorithms, third for comparison between manual and automatic results finally fourth one for conclusion.

Keywords*: Unstructured data, opinion mining, sentiment analysis, text mining, feature mining.*

1. Introduction

In recent time every person is busy and it is difficult to find time for shopping. Therefore, a new trend comes into existence which is known as online shopping. Shopping online saves our time and gives more choices. But before going to buy a particular product it is important to get information about that product from the one who bought it before. Instead of asking from others information can be extracted from the reviews attached with the products. These reviews help the one to take a decision over the product. But there are some problems with these reviews. One problem is big number of reviews and it is difficult to read all reviews. Second problem is free style of reviews. Because there is no pre defined format to write a review, anyone can write in own style. For example, one can use short words like gr8 for great, gd for good in the review. Similarly, one can write only one-line review or more than one lines review. Emotions can also be expressed like I am sad; I am happy etc. As a result, there is a need to automate a process to extract feature along with opinion from different reviews and provide a summary to the customer. This summary will help customer and seller both. Seller will know the popularity or weakness and competitor of product. Similarly, customer can take decision to buy that product. This problem to extract opinion along with attribute is known as opinion mining or sentiment analysis. Pang and Lee [1] and Bing Liu [2] presented two long surveys in which they focused on challenges, techniques and applications of sentiment analysis (SA). In [17] a prototype system named Opinion Observer is implemented. New trends in SA are presented beautifully in papers [3, 4, 5, 9]. In [18] authors presented adjective and adverb based sentiment analysis. Walaa [6]

ISSN: 1975-0080 IJMUE

also presented a paper on sophisticated categorization of a large number of recent articles just according the techniques. SA can be three levels one is document level where sentiment of whole article or topic is presented, second is sentence level where sentiment of a line is presented and third is aspects level where feature wise sentiments are presented. Techniques for SA are illustrated in the following figure 1.

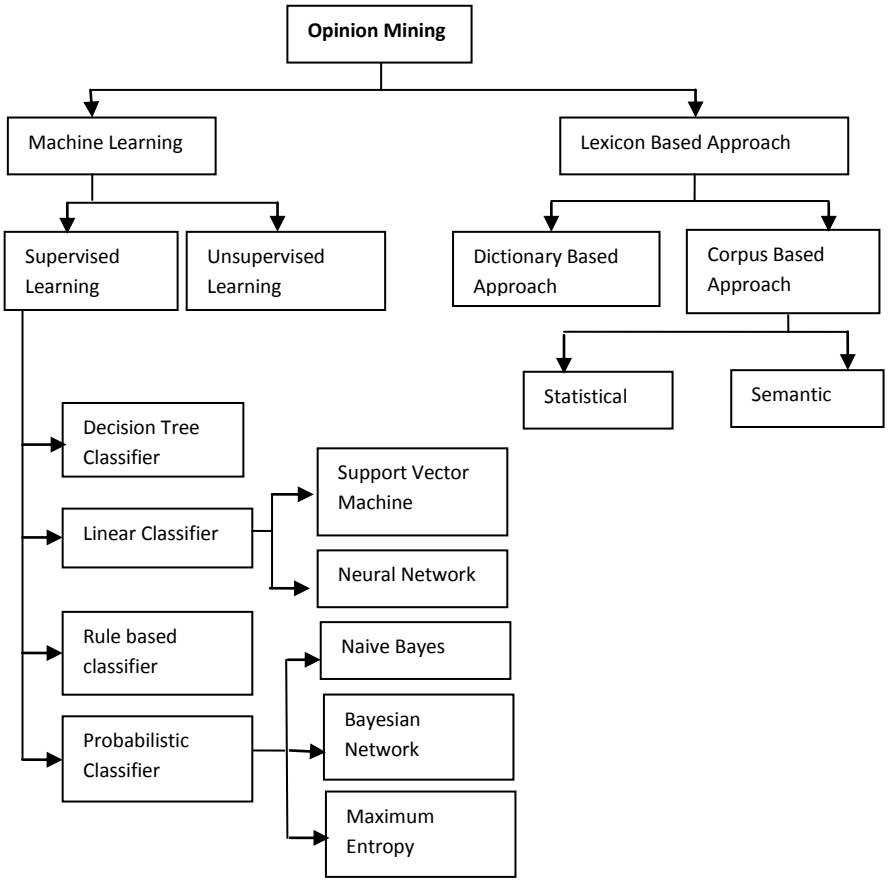

Figure 1. Different Techniques for Sentiment Classification

Data set for the analysis can be reviews of different products and services, news, articles about a topic, web pages, micro blogs and others. Data source for any data set may be any shopping site or it can be obtained from authors who are already working in this area. For example, Bing Liu who is already working in this area can provide data as per request. After getting knowledge on classification techniques and data set focus is just to implement algorithms.

2. Proposed Algorithms

Algorithm No.1:

This algorithm is lexicon [10, 11, 12, 13, 14] based where two list one for positive words and second for negative words are used. These two lists are referred from [8]. Architecture for this algorithm is shown in figure 2.

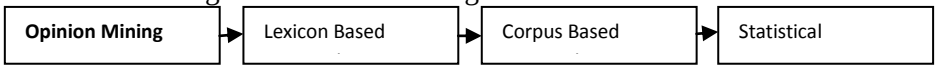

Figure 2. Lexicon based Architecture

If a new word is found means which is not available in these two lists, then this word and its closest words (synonyms) are collected from WordNet [7] which is a lexical database for English and added into the existing lists accordingly. For this purpose, an algorithm is developed and implemented in Java. The idea which is latest in the paper is detailed algorithms with Java code for the beginners' understanding and implementation.

Variables:
int cp – temporarily stores no. of positive words in a line
int cn - temporarily stores no. of negative words in a line.
int counter – It is used to know the positivity or negativity of word
int pcounter – It counts positivity for every line
int ncounter - It counts negativity for every line
double result
There are two sets one is for storing positive words (pos) and another one is for storing negative words (neg)
set <string> pos = new TreeSet < > ( );
set <string> neg = new TreeSet < > ( );
There are two scanners class objects in1 and in2
in1 - It is used to read words from file which contains positive words
in2 - It is used to read words from file which contains negative words
p_file – It is used to read from text file having reviews

Step 1: Read all words from the files which contains +ve and –ve words into pos set and neg set respectively

```
while(in1.hasNext() || in2.hasNext())
{
 if(in1.hasNext())
        pos.add(in1.next());
  if(in2.hasNext())
       neg.add(in2.next());
}
```

Step 2: Read the file line by line until file ends and count positive words and negative words for each line

```
Set counter = counter * 1;   // no change in case of positive word encountered
    counter  *=  (-1) ;  //  if  negative word encounters
```

Step 3: Reading a complete sentence check the value of counter

```
if(counter= = 1)
    pcount++;
else if (counter = = -1)
    ncount--;
```

Step 4: Repeat step 2 and 3 until end of file

Step 5: Now compare the pcount and ncount for the result

```
result = (pcount > ncount)? Pcount /(double)( pcount + ncount) : ncount / (double)(pcount + ncount)  ;
```

Step 6: Display result that may be positive or negative

```
if( pcount > ncount)
     System .out.println(" Result is" + result + "% Positive") ;
else if(ncount >  pcount)
```

System . out.println(“ Result is” + result + “% Negative”) ;
else
System .out.println(“ Result is neutral”) ;
Step 7:
End

This approach is applied on the data set canon.txt (reviews about digital camera) which contains more than 250 lines. Result comparison between manual and approach no. 1 is shown in the following table no.1

Table 1. Comparison between Manual vs. Approach No.1

Manual Process	**Approach No. 1**
No. of positive =162 No. of negative lines= 59 Result = $\frac{\text{Bigger no.}}{\text{Bigger no.} + \text{Smaller no.}} \times 100$ In this example Bigger No. Is positive line and Smaller No. is negative lines $\therefore$ Result = $\frac{162}{162+59} \times 100$ Result = 7**3.3%** Positive	No. of positive pcounts = 118 No. of ncounts = 59 $\therefore$ Result = $\frac{118}{118+59} \times 100$ Result =66.7 **%** Positive

Analysis: Results of these two processes are very close this may be due to effectiveness of this approach No. 1 or may be by chance. This approach is still to be applied and verified on other data sets to see the effectiveness. This approach is lexicon based means it considers only positive and negative words which may be relevant or not. Thus there is need of approach which provides summary of relevant opinion means feature based. So as a result the second approach is presented to achieve desire result.

Algorithm No.2:

This algorithm is based on opinions extracted according to the features[15,16]. In this frequent features are taken into consideration. Common architecture and detailed architectures are exhibited in figure 3 and 4 respectively.

Common Architecture:

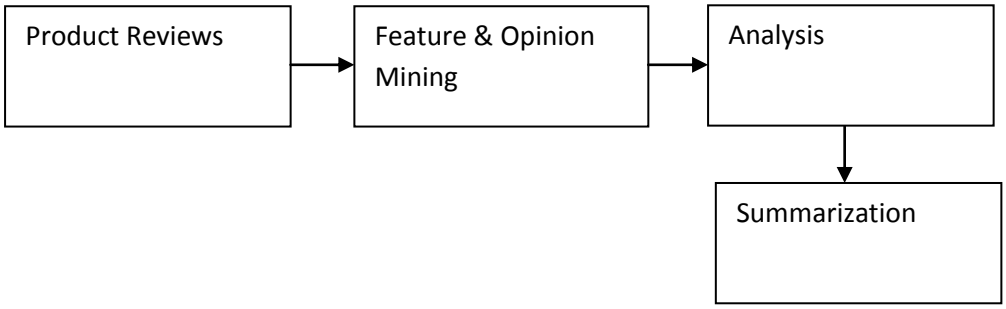

Figure 3. Feature based common Architecture

Detailed Architecture:

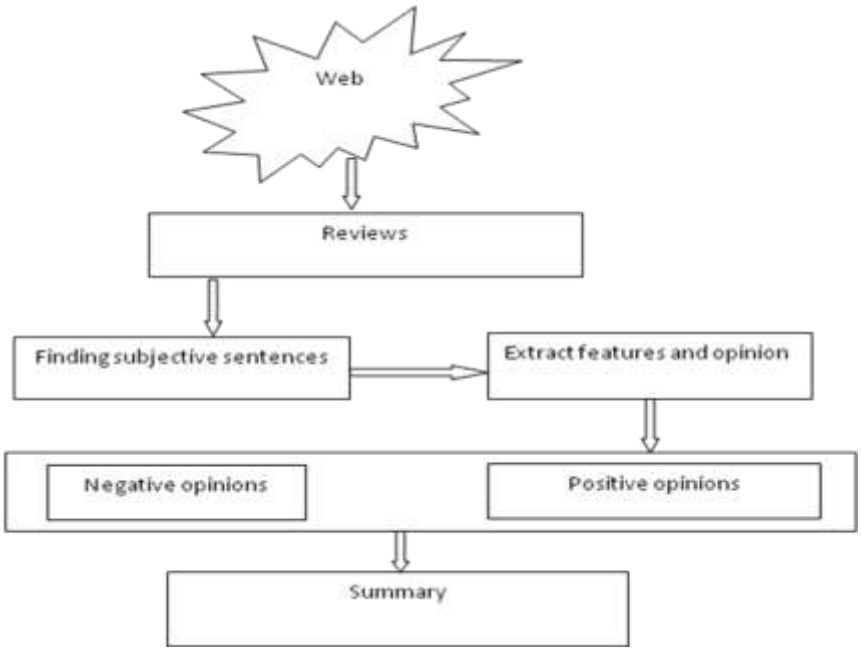

Figure 4. Feature based Common Architecture

I/PRP want/VBP to/To start/VB off/RP saying/VBG that/IN this/DT camera/NN is/VBZ small/JJ for/IN a/DT reason/NN ./.

A/DT larger/JJR memory/NN card/NN and/CC extra/JJ battery/NN are/VBP good/JJ things/NNS to/TO buy/VB ./.

Figure 5. Sample of Tag File as Target for Processing

In this approach three scanners class objects are used to read data from file having content as shown above in figure 5.

in – to read data from tag file
in1 – to read data from positive words [8,9] containing file
in2 – to read data from negative words [8,9] containing file
There are five lists as follows -
jj1 – to store adjectives of every line
nn – to store complete nouns in tag file
nn1 – to store nouns of a particular line
rb1 – to store adverb of a particular line
fnn – to store most frequently occurring nouns

One Tree Map is deployed to store nouns with their key values as their number of occurrences. Two set named pos and neg are maintained to store positive and negative words respectively.

```
Map < String, Integer>  a = new TreeMap <String, Integer> ( );
```

Step 1: All positive words and negative words are added into pos and neg lists respectively

```
Set <String> pos = new Tree Set< >( );
while(in1.hasNext( ) || in2.hasNext())
{
        if(in1.hasNext())
                pos.add(in1.next());
        if(in2.hasNext())
                neg.add(in2.next());
}
```

Step 2: Extract all nouns from the tag file

```
while(in.hasNextLine())
{
        String S = in.nextLine();
        S=  S.tolowerCase();
        S.trim();
                if(S.equals(“ “))
                continue;
int index =0;
        While(index >=0)
        {
        String tmp = “ “;
        int t= s.indexof(“/nn”,index);
        if(t<0)
        break;
        for(int  i= t-1; i >= 0 && S.CharAt(i) != ‘ ‘;i--)
                Tmp= S.CharAt(i) +tmp;
nn.add(tmp);
index = t+1;
        }
}
```

Step 3: Calculate the frequency of each noun from the aforesaid noun list

```
 for(int i= 1;i<nn.Size();i++)
{
```



```
        String spy = nn.get(i);
        if(a.containsKey(spy))
        {
        a.put(spy, a.get(spy) +1);
        }
        else
        {
        a.put(spy,1);
        }
}
```

Step 4: Set the threshold according to which nouns are extracted

$$\text{Threshold} = \frac{\text{Sum of frequecies of each word}}{\text{Total No. of Words}}$$

```
Set <String> eit = a. KeySet( );
for( String n:eit)
{       int temp3 = a.get(n);
        Sum1 =sum1 + temp3;
}
Threshold = (int)Math.Ciel((double)sum1/(double)(a.size());
```

Step 5: Extract the most frequently occurring nouns from the above nouns based on the threshold. Assuming nouns having frequency above threshold are known as most frequent nouns.

```
Set <String>  ei = a. KeySet( );
for(String n: ei)
        {
        int temp 2 = a.get(n);
                if(temp2 > Threshold -1)
                        fnn.add(n);
        }
```

Step 6: Now checking again from starting whether the sentence contains a noun which belongs to the list of most frequently occurring ones (fnn)

if it is true then continue to the next step

else search in next line and repeat this again until the file has been read completely.

Step 7: If noun belongs to the most frequently occurring list than find the all adjectives and adverbs in the same line and check whether these belong to positive words list or negative list.

Step 8: First check for adjectives one by one

If any adjective belongs to positive word list then there is no change. But in case of negative word counter variable is changed as follows

counter *= (-1);

Step 9: Pass same counter for rb's also

If rb be longs to the positive word list then there is no change

But in case of negative the counter will be multiplied by -1.

Step 10: Like the above way counter value can be 1 or -1 or 0.

If it is 1, then increment counterp by 1

If it is -1, then increment countern by 1

Step 11: Do steps 6 to 10 until read the whole tag file.

Step 12: Now calculate the result as follows

result = (counterp>countern) ?counterp / (double)(counter + countern) : countern / (double)(counterp + countern);

Step 13: Multiply result by 100 to get the actual percentage

if(counterp >countern) then

result is positive

and if(countern >counterp) then

result is negative

Step 14: End

3. Results Comparison

This algorithm is applied on the same tag file then result is 64 % which is less than manual approach and algorithm 1. This is due to relevant features wise opinions are extracted. Comparison between manual, algorithm 1 and 2 is shown in the following graph.

Table 2. Comparison between Manual, Algorithm #1 and Algorithm #2

Approach Type	Sentiment Percentage
Manual	**73.3**
Algorithm #1	**66.7**
Algorithm #2	**64**

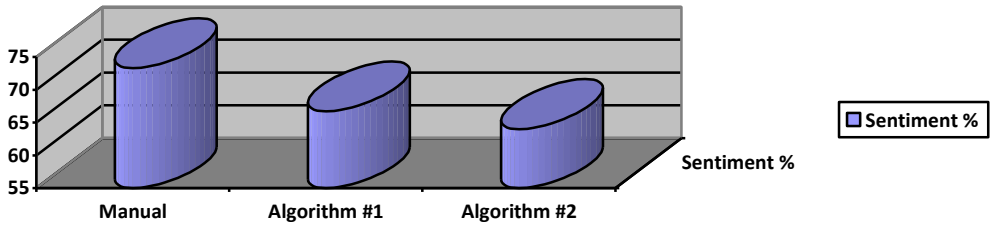

Figure 6. Feature based Common Architecture

Analysis: Above mentioned graph shows that algorithm 2 is better than algorithm 1 for a given data set because this is giving more relevant information. These two algorithms are still to be verified on other data sets.

Strength of aforesaid algorithms:

- Very simple to understand and implement and quite effective.
- Flexible in terms of frequency

Limitations: These produce some unwanted aspects.

APPLICATIONS AREAS: There are different areas where these algorithms can be used to produce summary for individual use. Few are as follows.

i. **Influence decision of customer:** It helps customers in their decision making before buying any product or availing any service. Customer knows product components wise and compares it with other contemporary product if any.

ii. **Quality wise improvement:** Company knows about the popularity of the product/service and if some negatives are there about product or its component.

iii. **Knowing Competition:** Sentiment of different users is analyzed to know any new trends or demand. If any feature is discussed several times then it can be said that discussed feature is very important and it can be improvised to adhere new customers.

iv. **In Recommendation System:** If there is negative about anything then same thing from other companies can be recommended. For example if a there is more negatives about battery backup then battery of other companies can be advertised to pull attention of the customer.

v. **Spam detection in reviews:** In which we check trust worthiness of reviews. If any review has more no of adjectives (may be positive or negative) then such review can lead to spam. If review is not about product and its components then it can also be a spam. Similarly if review is about brand not about product and its components then this can also be spam.

vi. **Emotion detection:** Emotions which can be any type of these love, joy, surprise, anger, sadness, and fear. Emotions are closely related to sentiments. Based on emotions any tweet, blog, news, forum entry can be polarized. For example I am enjoying its display. This sentence is having positive response. Similarly furious words and slang can also be used to know about sentiment.

4. Conclusion

This paper focusses on two algorithms regarding opinion extraction and their results with manual process. The aim of this paper is to provide detailed algorithms for knowledge purpose which no one has provided. Yet in future these algorithms can be improved and applied for better results. Semantic of each sentence can be understood and presented accordingly. Implicit sentences can also be focused to improve result. Abbreviated or short words can rationally be considered to improve results. Opinion words are not limited and fixed and identifying all such words from explicit and implicit sentences is still a challenging task. This problem can be explored and tackled smartly in future research.

References

[1] B. Pang, L. Lee, "Opinion mining and sentiment analysis", Found Trends Inform Retriev, vol. 2, **(2008)**, pp.1-135
[2] B. Liu, "Sentiment analysis and opinion mining", Synth Lect Human Lang Technol, **(2012)**.
[3] E. Cambria, B. Schuller, Y. Xia, C. Havasi, "New avenues in opinion mining and sentiment analysis", IEEE Intell Syst, vol. 28, **(2013),** pp.15–21.
[4] R. Feldman, "Techniques and applications for sentiment analysis", Commun ACM, vol. 56, **(2013)**, pp. 82–9.
[5] M. Andre´s, M.-B. Patricio, B. Alexandra, "Subjectivity and sentiment analysis: an overview of the current state of the area and envisaged developments", Decis Support Syst, vol. 53, **(2012)**, pp. 675–9.
[6] Walaa, Ahmad, Hoda, "Sentiment analysis algorithms and applicatons", Ain Shams Engineering Journal, vol. 5, **(2014)**, pp. 1093–1113.
[7] https://wordnet.princeton.edu
[8] https://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html
[9] M. Hu and B. Liu, "Mining and Summarizing Customer Reviews", Proceedings of the ACM SIGKDD International Conference on Knowledg, Discovery and Data Mining, **(2004)**; Seattle, USA.
[10] N. Alena, P. Helmut, I. Mitsuru, "Recognition of Affect, Judgment, and Appreciation in Text", Proceedings of the 23rd international conference on computational linguistics, Beijing, **(2010)**, pp. 806-14.
[11] B. Heerschop, F. Goossen, A. Hogenboom, F. Frasincar, U. Kaymak, F. de Jong, "Polarity Analysis of Texts using Discourse Structure", Presented at the 20th ACM Conference on Information and Knowledge Management, **(2011)**.
[12] A. Moreo, M. Romero, J.L. Castro, J.M. Zurita, "Lexicon-based comments-oriented news sentiment analyzer system", Expert System with Application, vol. 39, **(2012)**, pp. 9166–80.
[13] B. Alexandra, H. Jesu´s, M. Andre´, "Detecting implicit expressions of emotion in text: a comparative analysis", Decision Support System, vol. 53, **(2012)**, pp. 742–53.
[14] S.M. Mohammad, "From once upon a time to happily ever after:tracking emotions in mail and books", Decision Support System, vol. 53, **(2012)**, pp. 730–41.
[15] Q. Liu, Z. Gao, B. Liu and Y. Zhang, " Automated Rule Selection for Aspect Extraction in Opinion Mining", Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, **(2015)**.
[16] Q. Liu, B. Liu, Y. Zhang, "Improving Opinion Aspect Extraction using Semantic Similarity and Aspect Associations", Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI-16).
[17] B. Liu, M. Hu, J. Cheng, "Opinion Observer: Analyzing and Comparing Opinions on the Web", WWW, **(2005)**; Chiba, Japan.
[18] F. Benamara, C. Cesarano and D. Reforgiato, "Sentiment Analysis: Adjectives and Adverbs are better than Adjectives Alone", ICWSM, **(2007)**; Boulder, CO USA

