ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 12, December 2015

Design of Sentiment Analysis System for Hindi Content

Mukesh Yadav [1], Varunakshi Bhojane [2]

P.G. Student, Department of Computer Engineering, PIIT, New Panvel, University of Mumbai, Maharashtra, India[1]

Assistant Professor, Department of Computer Engineering, PIIT, New University of Mumbai, Maharashtra, India[2]

ABSTRACT: As the Internet is reaching to more people within the world, there is huge increase in Hindi content. There are more requirements to perform sentiment analysis for Hindi Language. Sentiment analysis is the process of finding out the emotional tone behind a series of words which is used to gain and understand the attitudes, opinions and emotions expressed from language. Sentiment Analysis involves capturing of user's behavior, likes and dislikes of an individual from the text. The work of most of the sentiment analysis system is to identify the sentiments express over an entity and classify it into either positive, negative or neutral sentiment. Our proposed system for sentiment analysis of Hindi health news uses own corpus to find the overall sentiment associated with the document; polarity of words in the review are extracted from corpus and then final aggregated polarity is calculated which can sum as either positive, negative or neutral. Neural Network is used to train the polarity words stored in database to make the processing faster.

KEYWORDS: Sentiment Analysis, Polarity, Natural Language Processing, Corpus, Machine Translation, Unicode.

I. INTRODUCTION

Sentiment Analysis comes under Natural Language Processing task which finds a person's opinion or feelings over a topic or object or entity. It analyzes personal emotions, feelings, attitude and opinion of a speaker or a writer over an object and find sentiments expressed by the person. Hindi is the fourth highest spoken language in the world. The web compared to past years is currently enriched with non English languages too. There exist very few systems which calculate sentiment associated with Hindi text as sentiment analysis is very difficult for Hindi language due many different complexity associated with Hindi text. Well annotated standard corpora are still not available for Hindi language. Hindi language lacks availability of efficient resources like parser and tagger which are essential for extracting sentiment. HindiSentiWordNet (HSWN) & a well known English SentiWordNet is available but consists of limited numbers of adjectives and adverbs, which still needs to improvement to achieve higher accuracy. There are many cases in which words can be used in multiple contexts and context dependent word mapping is still a difficult task, error prone and requires manual efforts to find accurate polarity of words.

A framework for performing sentiment analysis on mix Hindi language is presented in this paper. Related works done and past literature is discussed in section II. Scope of research is in section III. Proposed system and its architecture is shown in section IV. Conclusion in section V. Acknowledgements in section VI. In the end references & authors.

II. RELATED WORK

In this section we cite the relevant past literature of research work done in the field of sentiment analysis for Hindi language.

Authors developed a sentiment annotated corpora in the Hindi movie review domain in which there are three approaches: a) training a classifier on annotated Hindi corpus and using it to classify a new Hindi document; b) they translated the given document into English and use a classifier trained on standard English movie reviews to classify the document & detect the polarity of the translated document using a classifier in English, assuming polarity is not lost

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 12, December 2015

in translation; c) they developed a lexical resource called Hindi-SentiWordNet(H-SWN) and implement a majority score based strategy to classify the given document.[1]

Authors proposed "Hindi Subjective Lexicon : A Lexical Resource for Hindi Polarity Classification" in which they created lexicon using a graph based method. They explored how the synonym and antonym relations can be exploited using sample graph traversal to generate the subjectivity lexicon. Their proposed algorithm achieved approximately 79% accuracy.[2]

Authors developed a system "Sentiment Analysis in Hindi" by using Semi-Supervised approach to train Deep Belief Network on small percentage of labelled data (150) and assign polarity to unlabelled data. 300 sentences dataset stored in xml. Sentence level sentiment is adopted. Data pre-processing stage, Deep belief architecture divided in 2 stages: pre-training model & fine tuning step. They report 71% accuracy using DBN on English language, 76% using active deep learning and 64% on Hindi language using Deep Belief Network. They used movie reviews.[3]

Author proposed "A Hybrid Approach for Twitter Sentiment Analysis" in which they introduced an approach for automatically classifying the sentiment of 60,000 Twitter messages. These messages were classified as either positive or negative. A 3 stage hierarchical model is proposed by the author; first : labeling with emoticans; second : using predefined list of words with strong positive or negative sentiments; third : tokens are weighted based on subjectivity lexicon. Accuracy obtained without discourse by SentiWordNet was 64.694, proposed probability based method 71.116,SentiWordNet then probability based method 69.511, probability based method then SentiWordNet 71.83, Hybrid approach (SWN+Probability) 72.563. With discourse results obtained for the accuracy were; for SentiWordNet 66.052, proposed probability based method 71.625, SentiWordNet then probability based method 70.193, probability based method then SentiWordNet 73.35, Hybrid approach (SWN+Probability) 73.72. [4]

Authors proposed & implemented a method for improving HSWN named "Sentiment Analysis of Hindi Review based on Negation and Discourse Relation" in which770 reviews were taken as input. N-gram & POS-tagged N-gram approaches were also used. Fleiss kappa coefficient value 0.8092 &dataset has 104 words. Accuracy of 82.89% for positive reviews, 76.59% for negative and overall 80.21%.for improved HSWN + negation + Discourse.[5]

Authors proposed "Survey on Sentiment Analysis and Opinion Mining Techniques" which gave the process of Sentiment Analysis is divided into 5 steps : process of Sentiment Analysis for Text(Lexicon generation), Subjectivity detection, sentiment polarity detection using Network Overlap Technique, sentiment structurization, sentiment summarization-visualization-tracking. There are four approaches to predict the polarity of a word. In the First strategy; an interactive game is provided which identify the polarity of the words. In the Second strategy, a bi-lingual dictionary is developed for English and Indian Languages. In the third strategy, word net expansion is done using antonym and synonym relations. In the fourth approach, a pre-annotated corpus is used for learning.[6]

Authors proposed a system in which they used WSD algorithm (Sense disambiguation) which is used to find the correct sense of the word on a context. SVM is used. Cxv Term presence, vs term frequency, term position are described. SVM separate the 2 categories and build a wide gap which are root words and some are some of the affixes. Steps of algorithm contains: stop words removal, sorting single column word line and stemming performed on sorted list in which each word is compared with next 10 words assuming that minimum that 10 morphological variants is present in list. If word is present as substring in next word then the word is broken as substring + remaining characters of word. Substring is treated as root/base form and remaining characters are treated as affix.[7]

III. SCOPE OF RESEARCH

The proposed system is capable of searching English words in Hindi i.e. mix Hindi input file and replace English words which are present in the input document with Hindi words stored in UNICODE form. For this we need database for

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 12, December 2015

both the types of data i.e. mix original Hindi input text file and other one is manually translated words are stored in file. We create our own corpus consisting of all the polarity words from all the documents on health news. By using this corpus we will find out the polarity of the entire document. The document can either be positive or negative or neutral.

IV. PROPOSED SYSTEM

In this section we provide an approach in sentiment analysis. The architecture is provided in figure 1.

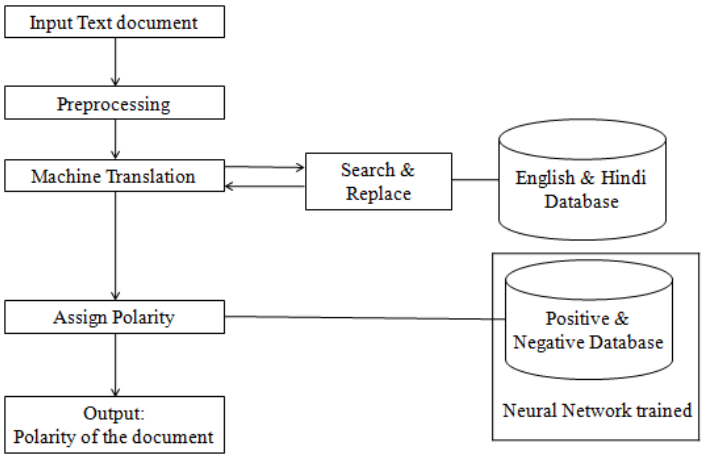

Figure 1 : Proposed Architecture

In figure 1, the proposed system takes input text document in Devanagri which is stored in a particular location. After selection of input file, pre-processing is done in which sentences are separated, words are separated, native to Unicode values are saved in matrix, special characters and symbols if present are removed. In next step, machine translation is done in which searching of English in Hindi words takes place. When an English word is found it is replaced with pure Hindi word. For this we need English database & Hindi database. The next step is assign polarity, in which we have two input: translated file & neural network trained polarity database and one output : polarity of the document. The last step in this proposed system is the output which is displayed to the user.

Let us consider an example shown in figure 2, in which we take input file containing data as " हेल्थ एक्सपर्ट्स के अनुसार स्विमिंग पूल में मिला क्लोरीन कई खतरनाक बीमारियों का कारण बन सकता है।" This is sent to next step pre-processing stage and stored in machine language i.e. in Unicode values. Next step is machine translation, in which we have English database in which original input paragraphs are stored line by line and in Hindi Database we have manually translated Hindi words for those English words. The system searches the values for which Unicode are not same, which means that there is English word present. Here, we have " हेल्थ", " एक्सपर्ट्स", "स्विमिंग" , "पूल", "क्लोरीन" are searched and replaced by these pure Hindi words: "स्वास्थ्य", " उस्ताद", "तैराकी", " तालाब", "नीरजी". Translated file is stored in a particular location. Then, the system calculates the number of polarity words using neural network trained dataset. In the end the output displayed is positive words equal to one, negative words equal to two, therefore polarity of the input file is negative.

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 12, December 2015

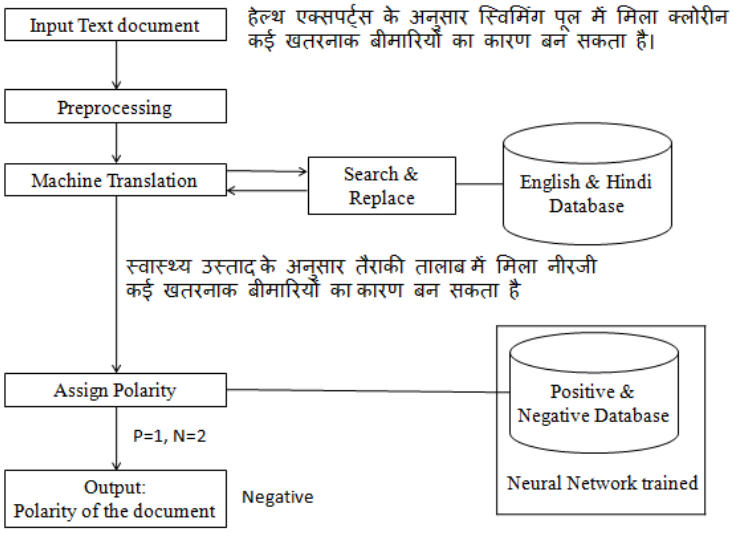

Figure 2 : Proposed Architecture Example

Steps in our proposed system

1. Take input file from the user.
2. Pre-processing
3. Machine Translation
4. Neural Network Training
5. Assign Polarity
6. Output displayed

Step 1 : Take Input file from the user

Select the input file from the user. The input text file must be stored in Unicode format only.

Step 2 : Preprocessing Stage

Preprocessing phase is used to convert the raw input into system acceptable format. Raw input is the set of strings or words which cannot be directly processed by the system. These sentences need to be normalized into the system specified format. As the proposed methodology uses the unicode of source input as one of the feature representation of raw input in Hindi needs to be done using the unicode format of the source language. In this, approach one character in English language is having single unicode and Hindi language is having two unicode.

Algorithm for Unicode generation

Input : Text file in Hindi language which contains reviews from blogs.

Output : Unicode for Hindi document for entire document, for each sentence, for each word, also for special characters.

1. Accept text file from the user
2. Read the file which is stored as unicode format.
3. Convert native language to unicode.
4. Arrange the content in an array.
5. Save the unicode for the entire document in a matrix say “ab.mat” which is our code_data.

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 12, December 2015

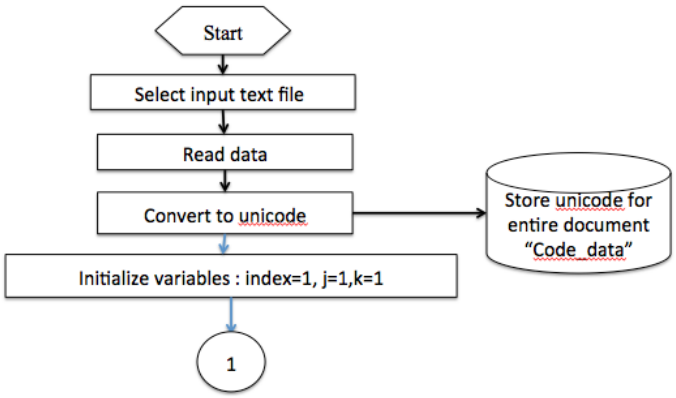

Figure 3 : Flow diagram for Unicode generation

In figure 3, it depicts flow diagram for Unicode generation for the input file selected. The process starts when we select input text file from the user. The data is read by the system and converted from native language to Unicode and stored in the matrix named "ab.mat" by the system which are used in further process. mat is the extension for storing MATLAB data.

Algorithm for Sentence Separation
Input : matrix for entire document i.e. ab.mat matrix
Output : Sentence matrix

1. Create a matrix 100 by 100 with zeros in them initially.
2. Initialize rows & columns. i.e. index=1, j=1;
3. for i=3 to end of the file i.e. ab matrix
4. if code_data(i)==46 && code_data(i+1)==0 //check presence of dot
5. then index = index + 1;
6. j-1;
7. cout(k)=code_data(i);
8. k=k+1;//if ends
9. else
10. separate(index,j)==code_data(i); //put data in matrix
11. j=j+1 // else ends
12. //for ends
13. save matrix for sentences in matrix "aa.mat"

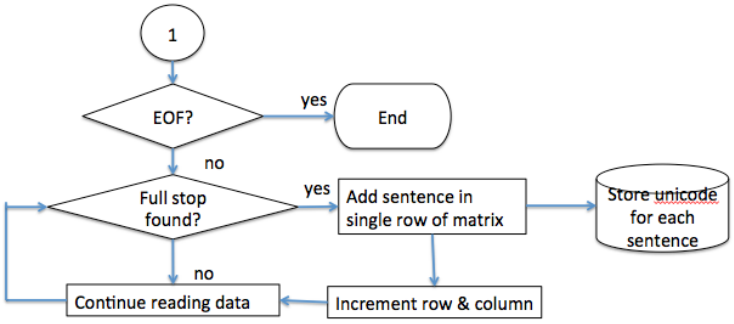

Figure 4 : Flow diagram for sentence separation

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 12, December 2015

In figure 4, it shows flow diagram for sentence separation in the input file. We search for full stop between two sentences in order to recognise end of a sentence. The Unicode for entire sentence is stored in a row and if a stop word is found then the Unicode storage for next sentence takes place in next row of the matrix. Our sentences are separated and stored in matrix named "aa.mat".

Algorithm for Word Separation
Input : Sentence matrix (aa.mat)
Output : Word matrix

1. Create matrix for word say 50 by 100.
2. Initialize rows and columns for word matrix i.e. ind=1, h=1.
3. for 1=1 to r // rows
4. for j=1 to c //columns
5. if separate(i, j)~=0 //checking space
6. if separate(i, j)~=32 //checking space
7. wordsep(ind, h)=separate(i, j);
8. h=h+1; //if ends
9. else
10. ind=ind+1;
11. h=1; //if ends
12. //for ends
13. //for ends
14. save the word separation matrix "bb.mat".

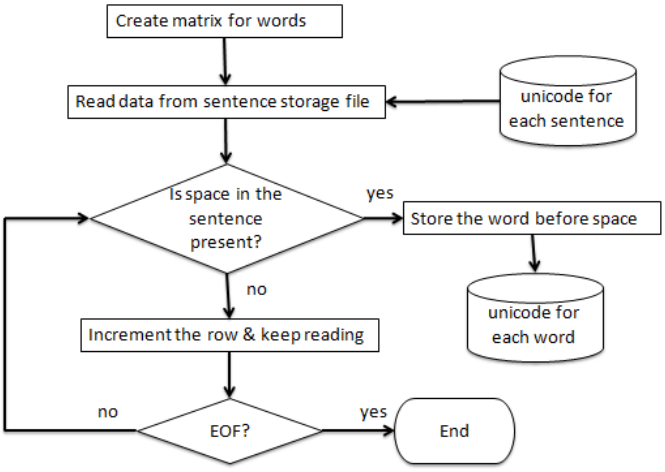

Figure 5 : Flow diagram for Word separation

In figure 5, it shows flow diagram for word separation for the words in the input file. Here we search for space between two words. If space is found then next words is stored in next line of matrix in database. This process is done till we reach the end of file where no words are present.

Removal of Special Characters
In this phase text is divided into a sequence of tokens, which are nothing but set of characters or words. Basic approach for tokenizing a text in our scenario; a sentence, is to separate characters if there is blank space between them and this separated set of characters becomes individual tokens. During this process also special characters which are

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 12, December 2015

encountered along with other characters are removed. The special characters such as " " ' ' , . / ?[ ] { } : ; \ | ~ ! @ # $ % ^ & * ( ) _ - = + <> are hence removed.

Step 3 : Machine Translation

In this step we have two database. One contains all the original paragraph words stored line by line excluding all the special characters. Another contains all the translated words done manually and all the other words in the paragraph except the non translated words from original paragraph. Both must be stored in the UTF-8 format only otherwise the system does not recognize these words and gives error.

Search & Replace

In this section we search for the Hindi word which is having English word in it. Example : लाइफ, हेल्थ, टेस्ट, रिजल्ट, पॉजिटिव etc.

Algorithm

1. Store input sentence in wordsep2
2. Store Hindi-in-English words in wordsep3
3. Store Hindi words in wordsep4
4. Compare wordsep2 & wordsep3
5. If match found then store the location of the word.
6. Then replace the word in wordsep2 with word in wordsep3.
7. Then send the sentence to polarity finding step.

Algorithm for searching

```
load Hindidb2    /*input by the user stored in Unicode is loaded*/
load dben   /* matrix of Hindi-in-English words are loaded Ex: लाइफ*/
for i=1:size(wordsep2,1)     /*for row i=1 till the end of wordsep2*/
for j=1:7  /*j is the column*/
 if wordsep2(i,j)~=0   /*if wordsep2 is not empty*/
wordd(i,j)=wordsep2(i,j); /*store the input matrix in another matrix say word(i,j) */
end end end /*close for, for & if loop*/
for i=1:size(wordsep3,1)  /* for row i=1 till the end of wordsep3*/
 for j=1:7  /*j is the column*/
 if wordsep3(i,j)~=0   /*if matrix is not empty*/
wordd2(i,j)=wordsep3(i,j);      /*store wordsep3 in wordd2 matrix*/
end end end /*close for, for & if loop*/
count=0;   k=1; /*count is no of matches, k is the location of the word to be replaced*/
[rr cc]=size(wordd2);  /*create an matrix with rr as rows & cc as columns*/
[r1 c1]=size(wordd);   /*create an matrix with r1 as rows & c1 as columns*/
for ii=1:r1    /*row is ii, starting from 1 till it reaches r1*/
for i=1:rr     /*row is 1, starting from 1 till it reaches rr*/
 count=0;  /*initialize a counter*/
 for j=1:cc   /*for column j=1 till it reaches cc i.e. end of the column*/
 if wordd(ii,1)~=0  /*if not empty*/
 if wordd(ii,j)==wordd2(i,j)      /*perform matching*/
 count=count+1;    /*increment the counter if match found*/
 end  end end  /*close for, if & if*/
  if count==cc     /*if match is found*/
  ind(k)=i;      /*store the match in ind(k) at position k*/
  k=k+1;     /*increment the counter if there are more matches*/
```

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 12, December 2015

27. end end end /*close if, if & if*/

OUTPUT: ind=5,6,7,8

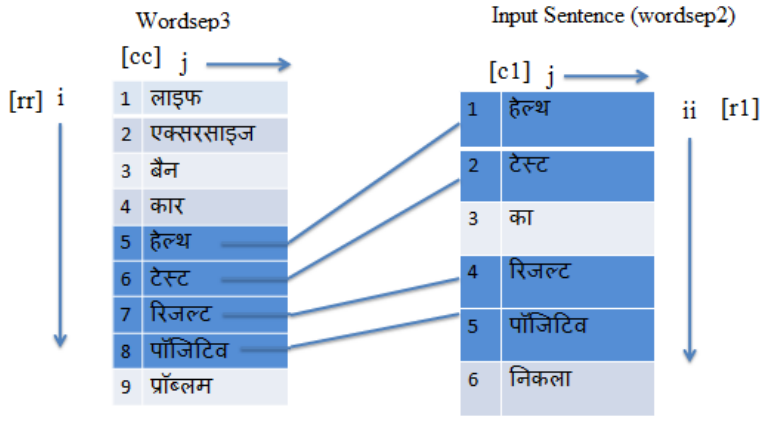

Figure 6 : Searching the matching word

In figure 6, it shows how searching is done to find out the English words in the input file. We have two matrix: wordsep3 & wordsep2. Wordsep3 contains all the English words in Hindi stored line by line & wordsep2 contains our input file words stored line by line. For each row & column, searching takes place. When a Unicode is found equal then the index is stored. Here, 5,6,7,8 will be stored. These indexes will be used for replacing the words.

Replace the word

In this section we replace the words with the match found in searching step with the pre stored database for the pure Hindi sentences. We have wordsep3 which contains mix Hindi words & wordsep4 contains pure Hindi words. The order & the index must be same in both the database.

Example : wordsep3(5,6,7,8)←→wordsep4(5,6,7,8) i.e. "हेल्थ", "टेस्ट", "रिजल्ट", "पॉजिटिव" must be replaced with "स्वास्थ्य", "जाँच", "नतीजा", "सकारात्मक".

Figure 7 : Replacing the word

In figure 7, it shows replacing of English words with pure Hindi words. In this process we need two database. English database in which all the English words are stored, here wordsep3, and Hindi database in which pure Hindi words are

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 12, December 2015

stored. The words in wordsep3 & wordsep4 must be equal, and each particular English word index must be same as its Hindi word index. Wordsep3 gave index as 5,6,7,8, so these will be same as in wordsep4. Now we need to replace the pure Hindi words in the input file. And give the output as translated file which contains only pure Hindi words.

Step 4 : Neural Network Training
Once our network has been structured, that network is ready to be trained. To start this process the initial weights are chosen randomly. Then, the training, or learning, begins.

Step 5 : Assign Polarity
In this step we need two input : translated file and the polarity file.

Polarity of the document Algorithm
1. If PosWords > NegWords then positive document.
2. Else if PosWords < NegWords then negative document.
3. Else neutral document

V. CONCLUSION

In existing system the Hindi input is in Devanagri script is taken to classify the words as positive, negative or neutral using HindiSentiWordnet which will have the polarity for all the Hindi words but it will not contain the mix Hindi words i.e. Hindi word in Devanagri but specifying English word in it. Example : "लाइफ" . Our proposed system will search English words in Hindi i.e. mix Hindi input file & replace English words which are present in the input document with Hindi words. Then the translated file containing translated words and the polarity file i.e. our corpus together will give the polarity of the document. The document can either be positive if positive words are greater than negative words, negative if negative words are more than positive words or neutral if both equal.

The use of neural network makes the processing faster. In future we will develop neural network and check if the polarity calculated is giving 100% accuracy or not and also can try different types of training algorithm and check which one gives faster output. This approach can be applied to the documents in mix Hindi format in various domains like decision making, business domain, social media monitoring etc which are in Hindi language. This will increase the database more and accuracy for different domain documents can be calculated and we can do sentiment analysis on more types of documents.

VI. ACKNOWLEDGEMENTS

I am using this opportunity to express my gratitude to thank all the people who contributed in some way to the work described in this paper. My deepest thanks to my project guide for giving timely inputs and giving me intellectual freedom of work. I express my thanks to the head of computer department and to the principal of Pillai Institute of Information Technology, New Panvel for extending his support.

REFERENCES

[1] Aditya Joshi,Balamurali AR, Pushpak Bhattacharya,"A Fall-back Strategy for Sentiment Analysis in Hindi: a Case study ", Proceedings of ICON 2010:8th International Conference on Natural Language Processing, Macmillan Publishers India.
[2] Akshat Bakliwal, Piyush Arora, Vasudeva Varma, "Hindi Subjective Lexicon : A Lexical Resource for Hindi Polarity Classification", Proceedings of the Eight International Conference on Language Resources & Evaluation(LREC), 2012
[3] Naman Bansal and Umair Z Ahmed, Advisor: Amitabha Mukherjee, "Sentiment Analysis in Hindi", Department of Computer Science and Engineering , Indian Institute of Technology, Kanpur, India, pg. 1-10, 2013.
[4] Namita Mittal et al., "A Hybrid approach for Twitter Sentiment analysis", http://ltrc.iiit.ac.in/icon/2013/proceedings/File27-paper58.

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 12, December 2015

[5] Namita Mittal, Bansat Agarwal, Garvit Chauhan, NItin Bania, Prateek Pareek, "Sentiment Analysis of Hindi review based on Negation and Discourse Realtion", International Joint Conference on Natural Language Processing, pg. 45-50,Nagoya, Japan,14-18 Oct 2013.
[6] Amandeep Kaur and Vishal Gupta, "A Survey on Sentiment Analysis and Opinion Mining Techniques", Journal of Emerging Technologies in Web intelligence, Vol. 5, No. 4, Nov 2013
[7] Sneha Mulatkar, "Sentiment Classification in Hindi", International Journal of Scientific & Technology research(IJSTR 2014), Vol.3 , Issue 5, May 2014

BIOGRAPHY

Mukesh Yadav is currently a post graduate student pursuing Masters in Computer Engineering at PIIT, New Panvel, and University of Mumbai, India. She has received her B.E in Computer Engineering from University of Mumbai. She is having 2 years of past experience in the field of teaching. Her areas of interest are Natural Language processing, Machine learning, Website development.

Varunakshi Bhojane is Assistant Professor in Computer Engineering Department at PIIT, New Panvel and University of Mumbai, India. She has received her M.Tech. in Computer Engineering from Dr. Babasaheb Ambedkar Technological University. She is having 13 years of experience in teaching. Her areas of interest are Algorithm, Machine Learning, Natural Language Processing.