Sentiment Analysis in a Resource Scarce Language: Hindi

Article *in* International Journal of Scientific and Engineering Research · September 2016

DOI: 10.14299/ijser.2016.09.005

4 authors, including:

Vandana Jha
University Visvesvaraya College of Engineering

P. Deepa Shenoy
University Visvesvaraya College of Engineering

Venugopal K R
University Visvesvaraya College of Engineering

ISSN 2229-5518

Sentiment Analysis in a Resource Scarce Language:Hindi

Vandana Jha, Manjunath N, P Deepa Shenoy and Venugopal K R
Department of Computer Science and Engineering
University Visvesvaraya College of Engineering, Bangalore University, Bangalore, India
Email: vjvandanajha@gmail.com

Abstract—A common human behavior is to take other's opinion before taking any decision. With the tremendous availability of documents which express opinions on different issues, the challenge arises to analyze it and produce useful knowledge from it. Many works in the area of Sentiment Analysis is available for English language. From last few years, opinion-rich resources are booming in other languages and hence there is a need to perform Sentiment Analysis in those languages. In this paper, a Sentiment Analysis in Hindi Language (SAHL) is proposed for reviews in movie domain. It performs 1) preprocessing like stopword removal and stemming on the input data, 2) subjectivity analysis on the preprocessed data, to remove objective sentences that are not contributing to opinion of the input data, 3) document level opinion mining for classification of the documents as positive and negative using two different methods: Machine learning technique and Lexicon based classification technique. We have used Naive Bayes Classifier, Support Vector Machine and Maximum Entropy techniques for Machine learning. In Lexicon based classification, adjectives are considered as opinion words and according to the polarity of the adjectives, the documents are classified, 4) negation handling with window size consideration for improving the accuracy of classification.

The effectiveness of the proposed approach is confirmed by extensive simulations performed on a large movie dataset.

Index Terms—Bollywood, Hindi, Natural Language Processing, Opinion Mining, Resource Scarce Language, Sentiment Analysis

1 INTRODUCTION

POSTING our opinions on the web has become extremely easy with Web 2.0. After watching movies or using any product or visiting some place, we can post movie reviews, product reviews or tourism related reviews. This opinion-rich data is of interest to the people in decision making about the entities in question and to the organizations for improving their products or services. Rather than media stars speaking on the behalf of general public, it gives the people a chance to express themselves. People get an opportunity to be heard by posting their viewpoint on web. That is the reason behind the availability of tremendous documents containing writer's viewpoint on the web. Now this is a challenge to mine meaningful information from those documents. This boosts usage of Sentiment Analysis or Opinion Mining.

"Sentiment Analysis (Opinion Mining) is the process of computationally identifying and categorizing opinions expressed in a piece of text, especially in order to determine whether the writer's attitude towards a particular topic, product, etc. is positive, negative, or neutral."

It is interdisciplinary and vibrant area of research in the domain of machine learning and text mining. It's intention is to unearth the viewpoint of a writer for finding opinion orientation with respect to a topic in the document. Hence, it is a combination of human intelligence and machine intelligence for text analysis and classifying the sentiments of user into positive, negative and neutral classes [1]. The word "sentiment analysis" and "opinion mining" is used interchangeably in this paper.

The popular and available opinion-rich contents are movie reviews, product reviews, blogs and posts. Sentiment Analysis can be performed at three levels: Document level, Sentence level and Aspect/Feature level. The polarity is determined for the overall document in Document level Sentiment Analysis. The polarity is decided for the individual sentences of the document in Sentence level Sentiment Analysis. The polarity is decided for the aspects/features of the document in Aspect level Sentiment Analysis.

Primary methods applied in Sentiment Analysis are:

- Using Subjective Lexicon - It is a database of words or phrases with a score assigned to each word. This score indicates the features associated with that word for its classification into positive, negative or neutral categories.
- N-Gram Modeling - It is the formation and use of a N-Gram model (unigram, bigram, trigram or combination of these) with given training data for categorization.
- Using Machine Learning - It makes prediction on data by obtaining the features from the text and performing supervised or semi-supervised learning.

1.1 Motivation

Many research works in Sentiment Analysis are available in English language. Only 28.6% of Internet users understand English[1] so it is essential to focus on Sentiment Analysis in other languages also. We are performing Sentiment Analysis for Hindi Movie Reviews. This is selected as dataset because huge amount of capital is invested on Bollywood movies. The year 2015 itself saw 204 releases with the cumulative net gross of over 27.25 billion rupees (US $425.78 million)[2]. Hindi, the 4th largest spoken language, has 310 million speakers across the world which is 4.45% of the world population and is the official language of India[3]. With the introduction of Unicode (UTF-8) standards, web pages in Hindi language have increased rapidly. But it is a difficult task because of the following challenges:

- Hindi is a resource scarce language. Absence of good Hindi language tagger and annotated corpus makes sentiment analysis a challenging task.
- Standard datasets are not available, which makes collection/creation of dataset a time consuming task.
- In the absence of standard dataset, comparison of techniques applied and results obtained, is a difficult task.

We have tried to overcome these challenges in some way and manage to mine Hindi dataset and extract the information out of it.

1.2 Contribution

In this paper, Sentiment Analysis in Hindi Language (SAHL) is proposed for movie reviews. A part of our work is published in [2]. Here, we extend on that work in several ways:

1) The dataset size, i.e., the number of files containing movie reviews is increased from 200 to 1000.
2) Preprocessing steps like stopword removal and stemming is performed on the input data.
   Real world data collected from the various internet sources may not be proper, hence the data needs to be polished and preprocessed before its use. Here, stopwords are removed from the initial acquired corpora.
   Next stemming is performed on the stopwords removed corpora.
3) Subjectivity analysis is performed on the preprocessed data.
4) We have used Naive Bayes Classifier, Multinomial Naive Bayes Classifier, Support Vector Machine and Maximum Entropy techniques for Machine learning methods. A comparative analysis is performed between the results obtained by different methods.

Collection of 1000 movie review dataset (500 positive and 500 negative files) and building a list of stopwords, both in Hindi language, for this work, is also our contribution and can be made available and utilized in future for research purposes only.

1.3 Organization

The organization of the paper is as follows. We first review related work in section 2. Our proposed work, SAHL is described in section 3. Simulations performed on real dataset obtained from various Hindi websites[4] and the results are discussed in section 4. The paper concludes in section 5.

2 Related Work

In the last few years, researches in the area of opinion mining and sentiment analysis have shown significant developments. Papers [3], [4], [5], [6] and [7] provides state-of-art survey on sentiment analysis/opinion mining and text mining. The works have been performed in different directions but we are only citing works in two directions here: Machine learning techniques and Lexicon based classification techniques.

2.1 Machine Learning Techniques

Machine Learning Techniques are mainly applied in supervised methods. Supervised methods use pre-existing/collected opinion corpora. Sentiment analysis could then be performed by applying popular text mining techniques, combining linguistic and statistic tools. These methods, first, automatically learn all types of linguistic features or attributes and then build a model for each corpus. This computed model is later used to classify the test corpus. Table 1 summarizes a few important researches in the area of sentiment analysis using machine learning classifiers like Naive Bayes (NB), Support Vector Machine (SVM), Maximum Entropy (ME).

2.2 Lexicon Based Classification Techniques

In Lexicon Based Classification Techniques, classification is performed by comparing the polarity of a given text with word lexicons whose polarities are known before their use and this determines the sentiment orientation of the documents. Adjectives are recognized as the most important source to express sentiment orientation in a document by many researchers [16], [17].

Many works have been done in opinion mining area in English language. High cost involved in creating corpora and lexical resources for a new language restricts building tools to mine opinion for those languages. Regardless of this condition, works in other languages are increasing: e.g.,

[1] http://www.internetworldstats.com/stats7.htm
[2] http://boxofficeindia.com/Details/art_detail/finalclassifications2014#.VPEVpfmUdnh
[3] http://en.wikipedia.org/wiki/List_of_languages_by_number_of_native_speakers

[4] http://bbc.co.uk/hindi
http://www.webdunia.com/

TABLE 1: Studies Related to Machine Learning Techniques for Sentiment Analysis in English Language

Author Citations	Techniques	Dataset and its size	Accuracy (%)
Pang et al. [8]	NB, ME, SVM	Movie reviews (IMDb)-700(+) and 700(-) reviews	77-82.9
Dave et al. [9]	NB, ME, SVM	Product reviews (Amazon)	88.9
Pang et al. [10]	NB, SVM	Movie reviews (IMDb)-1000(+) and 1000(-) reviews	86.4-87.2
Chen et al. [11]	NB, SVM, Decision Trees C4.5	Books Reviews (Amazon)-3,168 reviews	84.59
Boiy et al. [12]	Multinomial NB, ME, SVM	Movie reviews (IMDb)-1000(+) and 1000(-) reviews, Car reviews- 550(+) and 222(-) reviews	90.25
Annett and Kondrak [13]	NB, SVM, Decision Tree	Movie reviews (IMDb)-1000(+) and 1000(-) reviews	75-80
Ye et al. [14]	NB, SVM, Character based N-gram model	Travel blogs (travel.yahoo.com)-600(+) and 591(-) reviews	80.71-85.14
Xia et al. [15]	NB, ME, SVM, meta-classifier combination	Movie reviews (IMDb)-1000(+)and 1000(-) reviews, Product reviews (Amazon)	88.65

Chinese dataset is used in [18] and German dataset is used in [19].

Relatively less work is present for Indian languages. By using English-Bengali bilingual dictionary and publicly available English Sentiment Lexicons, Paper [20] recommended a computational approach for evolving SentiWordNet (Bengali). Paper [21] discussed four computational methods to predict the orientation of a word. An online intuitive game is implemented that recognizes the orientation of the words in their first approach. A bilingual WordNet development is done using synonym and antonym connections in their third approach. In their fourth approach, a pre-annotated corpus is considered for training. Ekman's six emotion classes (anger, disgust, fear, happy, sad and surprise) along with three types of intensities (high, general and low) are considered by Paper [22] for the process of labelling words.

By employing EnglishHindi Word Net Linking and English SentiWordNet, Joshi et al. [23] created H-SWN (Hindi-SentiWordNet). Kim and Hovy [24] presented a system that automatically identifies the people who hold opinions about a given topic and the sentiment of each opinion. Hindi WordNet and Hindi Subjective Lexicon are used by Narayan et.al. [25] for the recognition of orientation of adjectives and adverbs. Paper [26], implemented the classification of bi-polar nature, positive and negative. Bakliwal et al. [27] created Hindi lexicon by using a graph based method. An efficient method based on negation handling and discourse relation to identify the sentiments from Hindi content is developed by Namita Mittal et al. [28]. They included more opinion words into the existing Hindi SentiWordNet (HSWN) and developed an improved, annotated corpus for Hindi language. Their work realized nearly 80% accuracy for classification of reviews. Jha et al. [2] developed an opinion mining system in Hindi for Bollywood movie review data set. They achieved an overall accuracy of 87.1% for classifying positive and negative documents. Paper [29] performed sentence level subjectivity analysis. They achieved approximately 80% accuracy in classification on a parallel data set in English and Hindi having 71.4% agreement with human annotators. Jha et al. [30] proposed a sentiment aware dictionary in Hindi language for multi-domain data. Paper [31] proposed a stopword removal algorithm for Hindi Language which is based on a Deterministic Finite Automata (DFA). They achieved 99% accurate results. Paper [32] proposed a reputation system for evaluating trust among all good sellers of eBay website and able to rank the sellers efficiently.

3 Proposed Work

Fig. 1 illustrates the architecture and data flow model of the proposed work. It is divided into following phases:

Phase 1: Corpora Acquisition phase

Phase 2: Preprocessing phase

Phase 3: Polarity Detection using Machine Learning Techniques

Phase 4: Polarity Detection using Lexicon Based Classification Techniques

Phase 5: Negation handling

3.1 Corpora Acquisition phase

1) *Collection of Movie Reviews:* Here, we aim at fishing out movie reviews from the Web. There are lots of websites[5] available, containing movie reviews in Hindi. The movie reviews are crawled from http://hindi.webdunia.com/bollywoodmovie-review for this work. Same movie can be rated 2.5 or 3 at one website and 3 or 3.5 at another website, respectively.

[5]http://bbc.co.uk/hindi
http://www.webdunia.com/
http://www.raftaar.in/

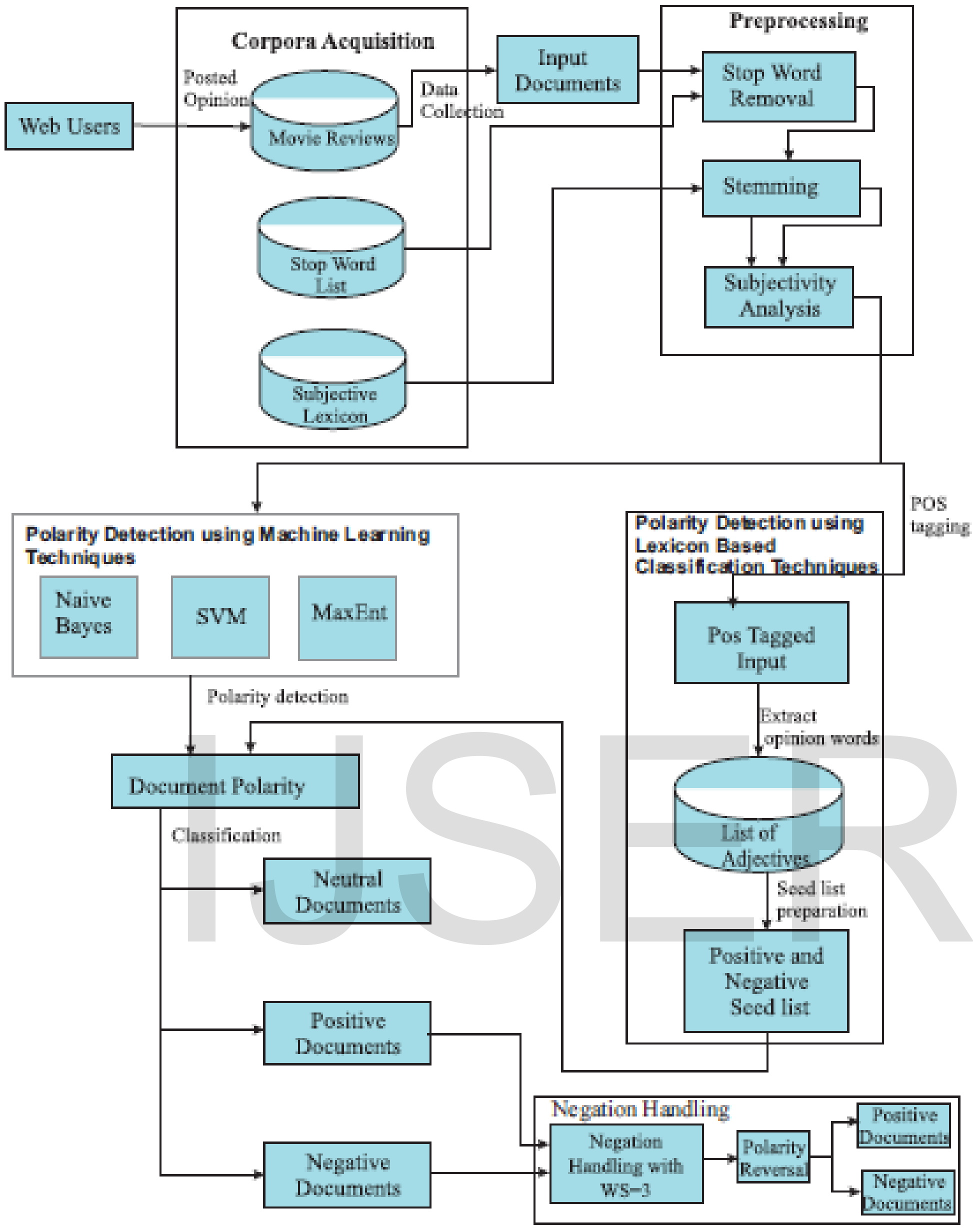


Fig. 1: Architecture of HOMS.

To avoid any inconsistency in rating the movies, the reviews are crawled from only one website. To avoid reviewer specific biasing, the reviews given by only designated reviewer are collected. The review ratings are based on 1-5 scale. On average, each movie review is 50 sentences long with 8 words in a sentence. A movie with more than 3 rating is considered as positive and less than 3 is considered as negative. A movie with rating 3 is assumed as neutral and discarded. The corpus is built in the similar manner as [33] into positive and negative classes. The dataset size is 1000 movie reviews (1000*50*8 = 400000 words), with 500 positive and 500 negative documents. These reviews are not randomly selected; these are collected as it is available. The dataset is still in the growing phase because movie reviews in Hindi language are appearing online recently.

2) *Creation of Stopwords list:* Stopwords are frequent, evenly distributed, function words in any document corpus which does not add any meaning to the text content. Information retrieval from the corpus is not getting affected by removal of these words. It has been proved that removing the stopwords reduces the document size to a considerable extent and saves time in text processing [34] in Natural Language Processing. There are two sources where hindi stopwords are available online. First is Kevin Bouge list of stopwords in various languages including Hindi[6]. Second is sarai.net list[7]. Third source can be translation of English Stopwords available in NLTK corpus into Hindi using translator[8]. In this paper, the Stopwords list is the extended list using all three resources and contains words as well as phrases. The combined list is verified by one native speaker of Hindi language and finalized after necessary corrections. For the first time, the phrases are also kept in the list because a word in present continuous verb form changes to a phrase when written in Hindi. For example, "Speaking" in English is "बोल रहा हूँ" in Hindi, where "बोल" is the verb and is stored and "रहा हूँ" is the stopword and removed. The list of stopwords are futher divided into list of four words, list of three words, list of two words and list of one word. For example, List of stopwords,
four : [किया जा रहा है,…] in English *being done*
three : [के बारे में,…] in English *regarding*
two : [से अधिक, के लिए,…] in English *above, for*
one : [मैं, मेरा,...] in English *I, Mine*

These four, three, two, one lists are used in different ways to remove stopwords, based on number of words it has. Stopwords like न, ना, नहीं (in English *no, not*) are not kept in stopword list because that is required in our work for negation handling and we do not want to filter it out in the form of stopwords. The stopwords list has total 265 words and phrases, where 1 phrase is in the list of four words, 3 phrases are in the list of three words, 17 phrases are in the list of two words and 244 words are in the list of one word.

3) *Creation of Subjectivity Lexicon:* We have used the subjectivity lexicon created in our previous work [29] by using English subjectivity lexicon from OpinionFinder and translating it using translator[8] as well as English-Hindi bilingual online dictionary[9]. The final Hindi subjectivity lexicon consists of 8226 words with both strong subjective type and weak subjective type. Table 2 shows a sample from Hindi lexicon along with their English original form.

[6]https://sites.google.com/site/kevinbouge/stopwords-lists
[7]http://mail.sarai.net/private/prc/Week-of-Mon-20080204/001656.html
[8]https://translate.google.co.in/
[9]http://www.shabdkosh.com/

3.2 Preprocessing phase

1) *Stopword Removal:* Stopword removal is a very important type of preprocessing technique in text processing because it can reduce the length of a document to 30-40%, without affecting its sentiments. In this paper, the stopwords are removed in the order of four words list, three words list, two words list and then one word list. This order of stopword removal is explained clearly using variable list n in the Function 1, where n is the number of words in the list. So far as our knowledge is concerned, this method is used for the first time for stopword removal. This remove more words (four, three, two) together at one time, instead of looking for each as one word stopword. This increases accuracy and (time) efficiency. When list of one word is removed as stopwords, some conditions are considered like stopwords with '|', '?' and ','. These are sentence delimiters in Hindi language and required to be preserved for subjectivity analysis.

It is challenging to differentiate between these two symbols, '|' and '|'. First symbol is the delimiter in Hindi language but in many documents, second symbol has been used. In the process of removing stopwords and retaining this symbol, all the review files should have the same symbol but in reality, it was not the case and consumed a large amount of time to identify this issue.

TABLE 2: A Sample of Hindi Subjectivity Lexica

English Word	Associated attributes	Hindi Word
luck	strongsubj, noun, positive	भाग्य
renunciation	strongsubj, noun, negative	सन्यास
bankrupt	weaksubj, adj, negative	दिवालिया
exclusively	weaksubj, adj, neutral	केवल
loot	strongsubj, verb, negative	लूटना
understand	strongsubj, verb, positive	जानना

2) *Stemming:* Stemming is the process of removal of the suffix of a word and reduces it to the root word. For example, study, studies, studying, all reduce to the root word study. This is a prerequisite step in text processing because it helps in getting correct frequency of the words in the document. Function 2 is used for stemming in our work and its principle is as follows: Suffix list is stored in the form of dictionary of 5, 4, 3, 2 and 1 suffix. For example,

5 : [◌ाएंगी, ◌ाएंगे, ◌ाऊंगी,…] with length five suffix
4 : [◌ाएंगी, ◌ाएंगा, ◌ाआ◌ेगी,...] with length four suffix
3 : [◌ा◌ेगे, ◌ाने, ◌ाना,…] with length three suffix
2 : [◌ाई, ◌ाए,◌ाने,…] with length two suffix
1 : [◌ो, ◌ु, ◌ू, ◌ी, ◌े,...] with length one suffix

Stemming is performed first for length five suffix, then for length four suffix and so on in the order of 5, 4, 3, 2, 1 and based on the concept given by Ramanathan and Rao in [35]

3) *Subjectivity Analysis:* The steps for this part is given in Function 3 and its principle is as follows: Hindi input file, Doc is first parsed at the sentence level and for each sentence, it is parsed at word level. When the word is matched with the word present in Hindi translated OpinionFinder dictionary then its word_type is checked. If it is strong subjective type, its strong_subj_words_count is maintained for the sentence. Similarly weak_subj_words_count is also maintained. If one strong subjective word occurs, the sentence is labeled as subjective sentence. For weak subjective words, sentences are labeled as subjective if its occurrence is two. Objective sentences are removed from the input file and only subjective sentences are retained to perform polarity detection in the next phase. At a particular time for checking objectivity, three consecutive sentences are considered together as previous, current and next sentence and if all three are objective, only current sentence is considered as objective and is removed. If in this set of three sentences, any sentence is subjective, sentences are retained. This process is applied to avoid the loss of weak subjective sentences.

3.3 Polarity Detection using Machine Learning Techniques

Here, four classifiers from Natural language toolkit (NLTK) is used for polarity detection. These are NB, Multinomial NB, SVM and ME. Different classifiers have been used to compare their performance on Hindi data and specifically these are selected among many classifiers because according to the related work studies, these classifiers work better for text mining and sentiment classification.

Naive Bayes: A NB classifier is used when the input dimensions are high and is based on Bayes' theorem. It is a text classification approach that assigns the class c to a given document d given in Eq. (1).

$$C^* = \text{argmax}_c P(c|d) \quad (1)$$

where P(c|d) is the probability of instance d being in class c.

Multinomial NB: Multinomial NB is a variant of NB and is based on NB algorithm for multinomially distributed data. It is used in text classification where the input data are represented as word vector counts. The distribution is parametrized by vectors $\theta_b = (\theta_{b1}, \ldots, \theta_{bn})$ for each class b, where n is the number of features in text classification and θ_{bi} is the probability $P(a_i \mid b)$ of feature i appearing in a sample belonging to class b.

The parameters θ_b is estimated by a smoothed version of maximum likelihood, i.e. relative frequency counting:

Function 1: Stopword Removal

```
Data: Movie Review Corpus M, Dictionary of Stop
      Words D_SW in list 4, 3, 2, 1
Result: Movie Review Corpus with all the Stop Words
        removed, M_S
begin
  Initialize:
  Clean_n=" ", Count=0, list_n=[]
  Perform:
  for n in 4, 3, 2 do
    for each file, F in M do
      for each word W in F do
        if Count==n then
          for each word, D_W in list_n of
          D_SW do
            if Clean_n==D_W then
              Remove Clean_n from F
            end
          end
          Reinitialize the variables
        else
          Count = Count+1
          Clean_n = Clean_n+W
        end
      end
    end
  end
  for each file, F in M do
    for each word W in F do
      if W is in list_1 of D_SW then
        Remove W from F
      else
        if last character of W is " I" OR "?"
        then
          if the remaining characters W [:-1]
          is in D_SW then
            Remove W [:-1] from F
          end
        end
        if last character of W is "," then
          if the remaining characters W [:-1]
          is in D_SW then
            Remove W from F
          else
            Remove W[-1] from F
          end
        end
      end
    end
  end
end
```

$$\hat{\theta}_{bi} = \frac{N_{bi} + \alpha}{N_b + \alpha n} \quad (2)$$

where $N_{bi} = \sum_{a \in T} a_i$ is the frequency of occurrence of feature i in a sample of class b in the training set T, and $N_b = \sum_{i=1}^{|T|} N_{bi}$ is the total count of all features for class b.

Function 2: Stemming

```
Data: Stopword removed document, M_S, dictionary of
      suffixes in 5,4,3,2,1 order in list L for stemming
Result: Document with all the word stemmed, Doc
begin
    Perform:
    Tokenize M_S and store it in Doc_words as a string
    of words
    for word in Doc_words do
        length_word = len(word) //single word in
        Doc_words
        for L in 5,4,3,2,1 do
            if length_word > len(L+1) then
                for each suffix in the resp order do
                    if len(suffix) > length_word then
                        return //Invalid (does not require
                        to check further)
                    end
                    if word[length_word−len(suffix):] in
                    suffix: then
                        Doc = length_word−len(suffix)
                    else
                        Doc=word
                    end
                end
            end
        end
        return Doc
    end
end
```

Function 3: Subjectivity Analysis

```
Data: Document with all the word stemmed, Doc and
      stemmed OpinionFinder dictionary
Result: Sentences labelled as Subjective or Objective
        and their count
begin
    Initialize:
    strong_subj_words_count = 0,
    weak_subj_words_count = 0;
    strong_subjective = [], weak_subjective = [],
    objective = True;
    Perform:
    Parse each sentence from Doc
    for each word in sentence do
        if word in dictionary then
            if wordtype in dictionary is strongsubj then
                strong_subj_words_count += 1
                if strong_subj_words_count > 0 then
                    objective = false
                end
            else
                if wordtype in dictionary is weaksubj
                then
                    weak_subj_words_count += 1
                    if weak_subj_words_count > 1 then
                        objective = false
                    end
                end
            end
        end
    end
    return objective
end
```


Support Vector Machine: SVM classifier constructs hyperplane in a multidimensional space which divides the input data into different class labels. It applies an iterative training algorithm to minimize an error function and constructs an optimal hyperplane. According to the type of the error function, SVM models can be classified into following groups:

- Classification SVM Type 1, C-SVM classification
- Classification SVM Type 2, nu-SVM classification

Maximum Entropy: ME classifier uses search-based optimization to find weights for the features that maximize the likelihood of the training data. The probability of class c given a document d and weights is

$$P(c|d,\lambda) = \text{def}\frac{\exp\sum_i \lambda_i f_i(c,d)}{\sum_{c' \in C} \exp\sum_i \lambda_i f_i(c',d)}$$

We have used Unigrams (Uni), Bigram (Big) word features for finding the accuracy of the system.

3.4 Polarity Detection using Lexicon Based Classification Techniques

Here, adjectives are considered as opinion-rich text and based on polarity of the available adjectives in the document, the document is classified. The principle of lexicon based classification techniques for polarity detection is as follows:

To find opinionated words in a movie review, Part-Of-Speech (POS) tagging is a required step. A POS Tagger is a NLP tool that parse the sentence and assigns tag to each word in the sentence [36]. For example, the sentence is, "यह फिल्म अच्छी है ।" (This film is good), POS tagger gives the output "यह/DEM फिल्म/NN अच्छी/JJ है/VM" where DEM is Demonstrative, NN is Noun, JJ is Adjective and VM is Verb-finite [37]. The POS tag JJ (adjective) is used to extract "अच्छी" (in this example) which is an opinionated word. We have used a statistical POS tagger, Trigrams'n'Tags (TnT) [38] and extracted adjectives from the documents. TnT tagger is based on Markov model and performs well on Hindi data. TnT Tagger is popular for its robustness and speed, however it initially loads lex and trigram files which take time to load. Once the loading is finished, we expect the tagger to be very fast. For each j in adj_extract set, if num be very fast. For each j in adj_extract set, if number of occurrence of j is more than the threshold value which is set to 10, it is added to most_frequent word list. These most_frequent Hindi movie domain words are rated by five human experts. According to their opinion for the orientation of the word, the word list is divided into positive and negative seed list words. We have created a positive and a negative seed list of fifteen words each with their known polarity. All the adjectives i.e. j in adj_extract set are matched with the initial words in the seed list. If the match occurs with positive seed list word (resp. negative), the positive count (resp. negative) is increased. If the adjective is not in the seed list and occurring more than the threshold value, we have incremented the positive seed list (resp. negative), after considering its polarity. The seed list is also incremented by adding synonyms of the initial seed list words. Human experts are used for knowing the polarity of most frequent words. Only the words with high inter-annotator agreement (= 0.9 and above) are added in the most frequent words. The incremented list has twenty five words each in positive and negative list. The documents are classified according to the polarity of the adjectives. If positive adjectives are more in the review, the review is classified as positive otherwise it is negative.

3.5 Negation handling

In this phase, we have performed negation handling with window size (WS) consideration. WS corresponds to the words prior to and after the word with tag "NEG". Once tag "NEG" is encountered, the sentence level polarity detection is performed. We have taken WS = 3 and extracted the words within this window. If the extracted words are positive adjectives (resp. negative), it is replaced by negative (resp. positive) seed list word. For example,
"यह फिल्म अच्छी नहीं है।" (This film is not good.)
converted to
"!यह !फिल्म !अच्छी नहीं !है।"
After negation handling
"!यह !फिल्म बुरा है।" (This film is nasty.)
After this we have repeated the steps of Function 4 and classified the document.

4 Performance Evaluation

In this section, we present results after conducting the simulations to validate our system. The performance of different machine learning approaches as well as lexicon based classifier is evaluated under the hypothesis that the labels assigned after considering the ratings given by reviewer on 1-5 scale (explained in section 3.1.1) are the accurate annotations for the classification.

4.1 Machine Learning Results

The proposed system is tested for performance analysis using the split ratio for selection of the training and test sets. The results are computed using 10-fold, that is, from 50% training data and 50% test data to 95% training data and 5% test data but on average the system is performing better with 75% of the data set as training data set and 25% as testing data set and these results are only shown in the paper. First, the results are computed, after conducting the experiments on 400 positive and 400 negative documents. Then for every 10 document increase (5 positive and 5 negative) the model is repeated to verify its results and there is significant increase in accuracy till 910 documents size (455 positive and 455 negative). After that there is insignificant increase in accuracy. Results also show significant improvement after preprocessing of the initial reviews, which is supporting already well known findings.

The final results on 1000 documents (500 positive and 500 negative) using Naive Bayes Classifier is shown in table III (a) and by using Multinomial Naive Bayes classifier is shown in table III (b). Fig. 2 shows the related graph for their performance matrics. It is clear from the results that the system is giving better accuracy using Bigram features than Unigram features. According to the study of the previous work, Multinomial NB should be performing better than NB classifier (as shown in table I) but this is not the case with Hindi language reviews as both are performing equally good with 100% accuracy using Bigram features after preprocessing of the reviews. If Unigram features are used, Multinomial NB is performing better than NB classifier but this accuracy is lesser than Bigram features accuracy.

The result using SVM classifier on 1000 documents for the same split ratio (75% of the data set as training data set and25% as testing data set) is given in table IV and the related graph is shown in Fig. 3. The results of SVM classifier are not improving after preprocessing and it is same as before preprocessing. Bigram features are giving better accuracy than Unigram features but the difference in results are minimal.

The results for ME Classifier is shown in table V and its graph is Fig. 4. ME Classifier is also giving better accuracy using Bigram features than Unigram features. After preprocessing of the reviews, the results are better than before preprocessing with a good difference for Unigram features but preprocessing does not matter for Bigram features.

Fig. 5 is the accuracy comparison between all the four classifiers using Unigram and Bigram features. It is clear from the results that all the classifiers are showing significant improvement in their accuracy after preprocessing except SVM classifier for both Unigrams and Bigrams. Bigram features are performing better than Unigram features for all the classifiers. According to our results, for Hindi data, SVM and ME classifiers are giving best results for Bigram features whereas NB and Multinomial NB are also equally good but after preprocessing.

4.2 Lexicon Based Classification Results

Initially, we constructed a seed list of 15 positive and 15 negative words. These are the most frequent adjectives in the movie review domain. A sample is shown in table VI. In the next step, we have maintained count of each adjective in each document and matched those adjectives with this seed list. The adjectives which are not matching with the seed list words and occurring more than threshold value are added in the seed list words according to its polarity. By incrementing the seed list words; final list has 25 words both in positive and negative list. This list is freezes at 25 counts because no other

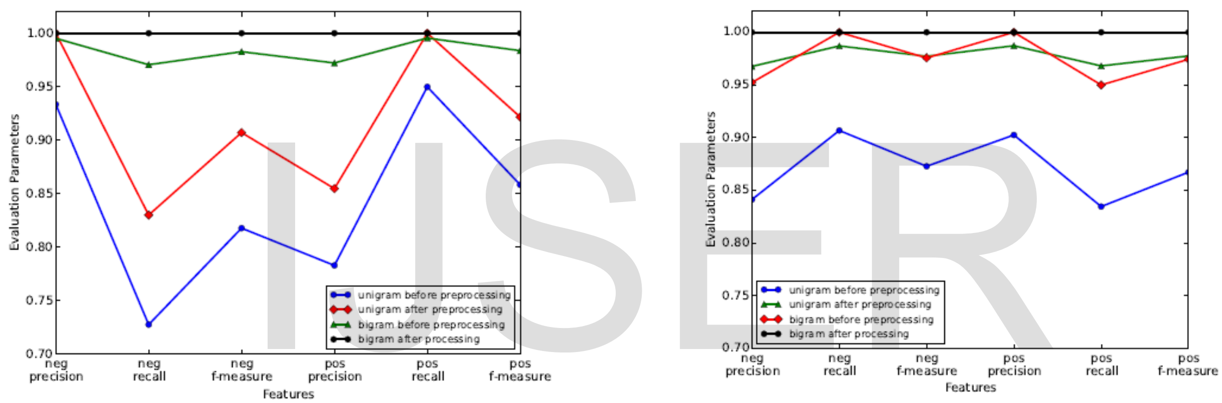
TABLE 3 : Fea stands for Features. NB and Multinomial NB Classifier is used to classify 1000 documents into Negative (Neg) and Positive (Pos) class using Unigrams (uni) and Bigram (big) features. The results in the form of accuracy percentage (A%), Precision (P), Recall (R) and F-measure (FM) is shown for both before and after preprocessing.

	Before Preprocessing							After Preprocessing						
Fea	Pos				Neg			Pos				Neg		
	A%	P	R	FM	P	R	FM	A%	P	R	FM	P	R	FM
Uni	84.05	0.933	0.728	0.818	0.783	0.949	0.858	91.5	1	0.83	0.907	0.855	1	0.922
Big	98.31	0.995	0.970	0.983	0.972	0.995	0.984	100	1	1	1	1	1	1

(a) Results of Naive Bayes Classifier on 1000 documents

	Before Preprocessing							After Preprocessing						
Fea	Pos				Neg			Pos				Neg		
	A%	P	R	FM	P	R	FM	A%	P	R	FM	P	R	FM
Uni	87.01	0.841	0.907	0.873	0.903	0.835	0.867	97.5	0.952	1	0.976	1	0.95	0.974
Big	97.75	0.968	0.987	0.977	0.987	0.968	0.977	100	1	1	1	1	1	1

(b) Results of Multinomial Naive Bayes Classifier on 1000 documents



(a) Naive Bayes Classifier

(b) Multinomial Naive Bayes Classifier

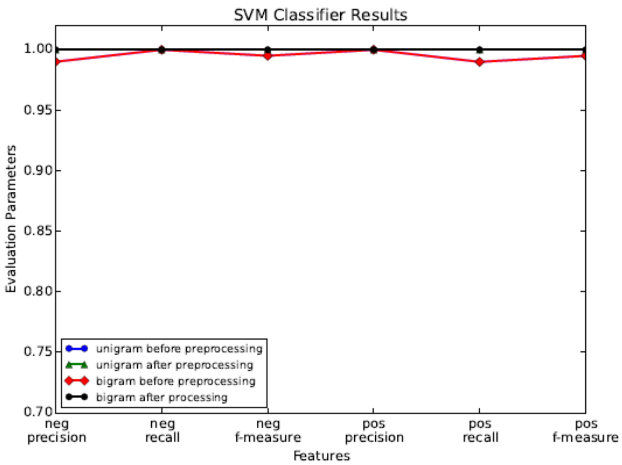
Fig. 2: NB and Multinomial NB Classifier Results on 1000 Documents

TABLE 4: Results of SVM Classifier on 1000 documents

	Before Preprocessing							After Preprocessing						
Fea	Pos				Neg			Pos				Neg		
	A%	P	R	FM	P	R	FM	A%	P	R	FM	P	R	FM
Uni	99.5	0.99	1	0.995	1	0.99	0.995	99.5	0.99	1	0.995	1	0.99	0.995
Big	100	1	1	1	1	1	1	100	1	1	1	1	1	1

TABLE 5: Results of ME Classifier on 1000 documents

	Before Preprocessing							After Preprocessing						
Fea	Pos				Neg			Pos				Neg		
	A%	P	R	FM	P	R	FM	A%	P	R	FM	P	R	FM
Uni	86.4	0.864	0.858	0.861	0.864	0.869	0.866	100	1	1	1	1	1	1
Big	100	1	1	1	1	1	1	100	1	1	1	1	1	1



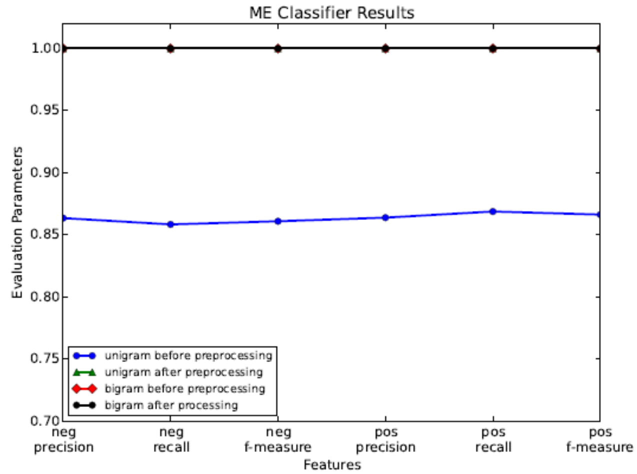


Fig. 3: Support Vector Machine Classifier Results on 1000 Documents

Fig. 4: ME Classifier Results on 1000 documents

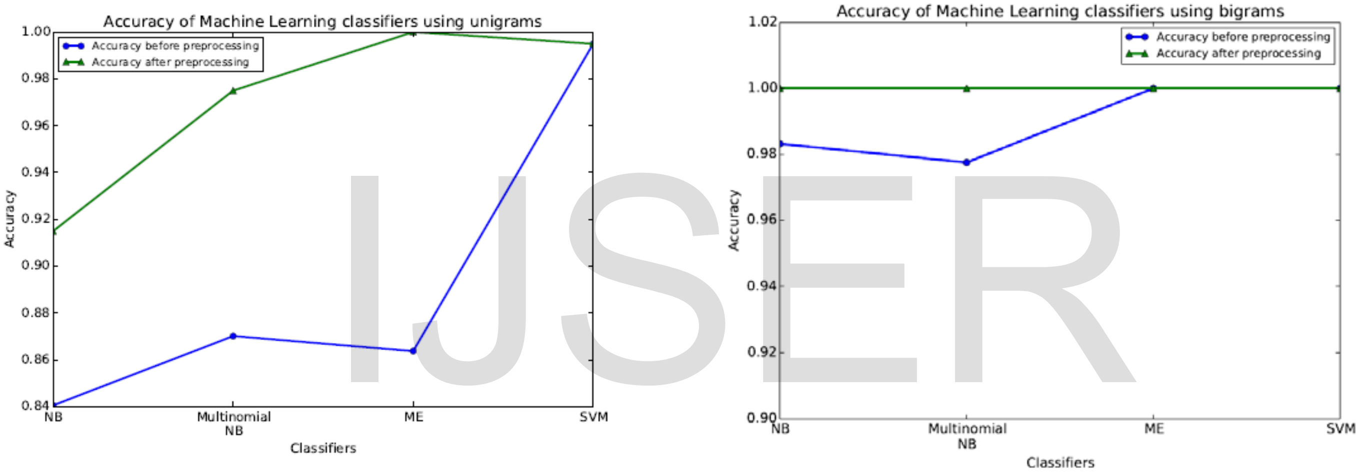


Fig. 5: Result Accuracy of Different Classifiers on 1000 Documents

TABLE 6: A Sample of Positive and Negative Seed list

Positive Seed list words(translation in english)	अच्छा (good), श्रेष्ठ (dominate), सर्वश्रेष्ट (best), उत्तम (best), सफल (successfull), बेहतर (better), सकारात्मक (positive), शिष्ट (well-mannered), सही (right)
Negative Seed list words(translation in english)	अश्लील (obscene), घटिया (poor), बेकार (useless), गलत (wrong), कमजोर (weak), बुरा (nasty), असफल (unsuccessful), नकारात्मक ((negative), लचर (poor)

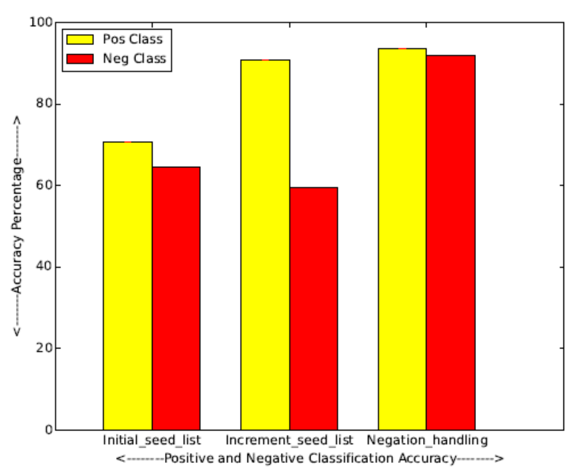


Fig. 6: Lexicon Based Classification Results

adjective is occurring more than threshold value in movie review domain except these. We have used both these lists, list of 15 seed words and list of 25 seed words, for polarity detection. The results for classifying 1000 documents as positive and negative using these two lists are shown in table VII and the generated graph is shown in Fig. 6. Accuracy is computed under the assumption that reviews classified as positive and negative using reviewer rating from http://hindi.webdunia.com/bollywoodmovie-review are accurate. With the initial seed list, accuracy is low. The incremented seed list has increased the accuracy of positive classification by almost 20% but decreased for negative classification. This is mainly because of the presence of sentences with positive adjectives preceded/followed with 'NEG' tag words like न, ना, नहीं (in English no, not). This is taken care in negation handling and after that step, the accuracy has increased remarkably.

TABLE VII: Accuracy for classifying 1000 documents using Lexicon Based Classifier

	Positive	Negative
Initial seed list of 15 words	70.73%	64.6%
Incremented seed list of 25 words	90.74%	59.39%
After Negation Handling	93.59%	91.92%

The accuracy percentage for classification of Hindi movie reviews into positive and negative classes is higher by Machine learning techniques than by Lexicon based techniques but Lexicon based techniques is much more transparent and the results can be checked/compared at any inbetween and final stages of processing. If some positive reviews are classified into negative and same number of negative reviews are classified as positive, it can be detected easily by Lexicon based techniques whereas detecting this situation is difficult by Machine learning techniques.

5 Conclusions

In this paper, we have proposed a method to determine the opinion orientation i.e. polarity of the Hindi movie reviews. There is a need for sentiment analysis in Hindi language because of the surge in Hindi data on the web. We have used NB classifier, SVM and ME in Machine Learning and Lexicon Based Classification Techniques to detect polarity of the documents. Simulation results show that our approach is performing well in the domain. We are performing many text mining approaches like stopword removal, stemming and subjectivity analysis to minimize noisy text and to improve accuracy. Future works may be manifold. First, our methods is not having very large database of movie reviews but we are increasing it on monthly basis as the new movie reviews are available. Second, in this work we focused on adjectives POS tag, we would also like to enhance the extraction task to other POS tag types. Third, our method is able to handle negative sentences and can also be extended to handle discourse relation like बल्कि (but rather), लेकिन (but). For example, कहानी अच्छी तो नहीं है लेकिन संगीत उम्दा है । (The story is not so good, but the music is great). This type of discourse relation is able to change the orientation of the sentence and can be considered as part of a future work.

References

[1] B. Pang and L. Lee, "Opinion mining and sentiment analysis," Foundations and trends in information retrieval, vol. 2, no. 1-2, pp. 1–135, 2008.

[2] V. Jha, N. Manjunath, P. D. Shenoy, K. Venugopal, and L. Patnaik, "Homs: Hindi opinion mining system," in Recent Trends in Information Systems (ReTIS), 2015 IEEE 2nd International Conference on. IEEE, 2015, pp. 366–371.

[3] B. Liu and L. Zhang, A Survey of Opinion Mining and Sentiment Analysis. Boston, MA: Springer US, 2012, pp. 415–463.

[4] J. Serrano-Guerrero, J. A. Olivas, F. P. Romero, and E. Herrera-Viedma, "Sentiment analysis: a review and comparative analysis of web services," Information Sciences, vol. 311, pp. 18–38, 2015.

[5] A. Kumar and M. S. Teeja, "Sentiment analysis: A perspective on its past, present and future," International Journal of Intelligent Systems and Applications, vol. 4, no. 10, p. 1, 2012.

[6] A. K. Nassirtoussi, S. Aghabozorgi, T. Y. Wah, and D. C. L. Ngo, "Text mining for market prediction: A systematic review," Expert Systems with Applications, vol. 41, no. 16, pp. 7653–7670, 2014.

[7] E. Cambria, B. Schuller, Y. Xia, and C. Havasi, "New avenues in opinion mining and sentiment analysis," IEEE Intelligent Systems, vol. 28, no. 2, pp. 15–21, 2013.

[8] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up?" sentiment classification using machine learning techniques," in Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10. Association for Computational Linguistics, 2002, pp. 79–86.

[9] K. Dave, S. Lawrence, and D. M. Pennock, "Mining the peanut gallery: Opinion extraction and semantic classification of product reviews," in Proceedings of the 12th international conference on World Wide Web. ACM, 2003, pp. 519–528.

[10] B. Pang and L. Lee, "A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts," in Proceedings of the 42nd annual meeting on Association for Computational Linguistics. Association for Computational Linguistics, 2004, p. 271.

[11] C. Chen, F. Ibekwe-SanJuan, E. SanJuan, and C. Weaver, "Visual analysis of conflicting opinions," in Visual Analytics Science And Technology, 2006 IEEE Symposium On. IEEE, 2006, pp. 59–66.

[12] E. Boiy, P. Hens, K. Deschacht, and M.-F. Moens, "Automatic sentiment analysis in on-line text," in ELPUB, 2007, pp. 349–360.

[13] M. Annett and G. Kondrak, "A comparison of sentiment analysis techniques: Polarizing movie blogs," in Advances in artificial intelligence. Springer, 2008, pp. 25–35.

[14] Q. Ye, Z. Zhang, and R. Law, "Sentiment classification of online reviews to travel destinations by supervised machine learning approaches," Expert Systems with Applications, vol. 36, no. 3, pp. 6527–6535, 2009.

[15] R. Xia, C. Zong, and S. Li, "Ensemble of feature sets and classification algorithms for sentiment classification," Information Sciences, vol. 181, no. 6, pp. 1138–1152, 2011.

[16] V. Hatzivassiloglou and K. R. McKeown, "Predicting the semantic orientation of adjectives," in Proceedings of the 35th annual meeting of the association for computational linguistics and eighth conference of the european chapter of the association for computational linguistics. Association for Computational Linguistics, 1997, pp. 174–181.

[17] P. D. Turney, "Thumbs up or thumbs down? semantic orientation applied to unsupervised classification of reviews," in Proceedings of the 40th annual meeting on association for computational linguistics. Association for Computational Linguistics, 2002, pp. 417–424.

[18] Y. Hu, J. Duan, X. Chen, B. Pei, and R. Lu, "A new method for sentiment classification in text retrieval," in Natural Language Processing–IJCNLP 2005. Springer, 2005, pp. 1–9.

[19] S.-M. Kim and E. Hovy, "Identifying and analyzing judgment opinions," in Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics. Association for Computational Linguistics, 2006, pp. 200–207.

[20] A. Das and S. Bandyopadhyay, "Sentiwordnet for bangla," Knowledge Sharing Event-4: Task, vol. 2, 2010.

[21] A. Das and S. Bandyopadhyay, "Sentiwordnet for indian languages," Asian Federation for Natural Language Processing, China, pp. 56–63, 2010.

[22] D. Das and S. Bandyopadhyay, "Labeling emotion in bengali blog corpus–a fine grained tagging at sentence level," in Proceedings of the 8th Workshop on Asian Language Resources, 2010, p. 47.

[23] A. Joshi, A. Balamurali, and P. Bhattacharyya, "A fall-back strategy for sentiment analysis in hindi: a case study," Proceedings of the 8th ICON, 2010.

[24] S.-M. Kim and E. Hovy, "Determining the sentiment of opinions," in Proceedings of the 20th international conference on Computational Linguistics. Association for Computational Linguistics, 2004, p. 1367.

[25] D. Narayan, D. Chakrabarti, P. Pande, and P. Bhattacharyya, "An experience in building the indo wordnet-a wordnet for hindi," in First International Conference on Global WordNet, Mysore, India, 2002.

[26] D. Rao and D. Ravichandran, "Semi-supervised polarity lexicon induction," in Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics. Association for Computational Linguistics, 2009, pp. 675–682.

[27] A. Bakliwal, P. Arora, and V. Varma, "Hindi subjective lexicon: A lexical resource for hindi polarity classification," in Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC), 2012.

[28] N. Mittal, B. Agarwal, G. Chouhan, N. Bania, and P. Pareek, "Sentiment analysis of hindi review based on negation and discourse relation," in Sixth International Joint Conference on Natural Language Processing, 2013, p. 45.

[29] V. Jha, N. Manjunath, P. D. Shenoy, and K. Venugopal, "Hsas: Hindi subjectivity analysis system," in 2015 Annual IEEE India Conference (INDICON). IEEE, 2015, pp. 1–6.

[30] V. Jha, R. Savitha, S. S. Hebbar, P. D. Shenoy, and K. Venugopal, "Hmdsad: Hindi multi-domain sentiment aware dictionary," in 2015 International Conference on Computing and Network Communications (CoCoNet). IEEE, 2015, pp. 241–247.

[31] V. Jha, N. Manjunath, P. D. Shenoy, and K. Venugopal, "Hsra: Hindi stopword removal algorithm," in 2016 IEEE International Conference on Microelectronics, Computing and Communications (MicroCom 2016). National Institute of Technology Durgapur, India: IEEE, 2016.

[32] V. Jha, R. Savitha, P. D. Shenoy, and K. Venugopal, "Reputation system: Evaluating reputation among all good sellers," in Proceedings of NAACL-HLT, 2016, pp. 115–121.

[33] (2004) Movie review data set. [Online]. Available: http://www.cs.cornell.edu/people/pabo/movie-review-data/

[34] B. F. William and R. Baeza-Yates, "Information retrieval: Data structures and algorithms," ISBN-10, vol. 134638379, 1992.

[35] A. Ramanathan and D. D. Rao, "A lightweight stemmer for hindi," in the Proceedings of EACL, 2003.

[36] C. D. Manning, "Part-of-speech tagging from 97% to 100%: is it time for some linguistics?" in Computational Linguistics and Intelligent Text Processing. Springer, 2011, pp. 171–189.

[37] (2006, November) Pos tag set for indian languages - ltrc - iiit hyderabad. [Online]. Available: http://ltrc.iiit.ac.in/nlptools2010/files/documents/POS-Tag-List.pdf

[38] T. Brants, "Tnt: a statistical part-of-speech tagger," in Proceedings of the sixth conference on Applied natural language processing. Association for Computational Linguistics, 2000, pp. 224–231.

Vandana Jha obtained her Bachelor of Engineering in Computer Science and Engineering from Maharshi Dayanand University, Gurgaon, India in 2003. She received her Masters of Technology specialized in the field of Computer Science and Engineering from Kuvempu University, Karnataka, India in 2009. Currently she is working as Research Scholar in the Department of Computer Science and Engineering, University Visvesvaraya College of Engineering, Bangalore University, Bangalore, India. Her research interests include Information Retrieval, Data Mining, Opinion Mining and Web Mining.

Manjunath Gouda received Bachelor of engineering from Visvesvaraya Technological University and Masters of Engineering from Bangalore university. He has done work in Natural Language Processing. His research interests include Data Mining, Text Analytic and Big Data Analysis. Currently he is working in Harman International (India) Pvt. Ltd.

P Deepa Shenoy is currently working as Professor in the Department of Computer Science and Engineering, University Visvesvaraya College of Engineering, Bangalore University, Bangalore, India. She did her doctorate in the area of Data Mining from Bangalore University in the year 2005. Her areas of research include Data Mining, Soft Computing, Biometrics and Social Media Analysis. She has published more than 150 papers in refereed International Conferences and Journals.

K R Venugopal is currently the Principal, University Visvesvaraya College of Engineering, Bangalore University, Bangalore. He obtained his Bachelor of Engineering from University Visvesvaraya College of Engineering. He received his Masters degree in Computer Science and Automation from Indian Institute of Science Bangalore. He was awarded Ph.D. in Economics from Bangalore University and Ph.D. in Computer Science from Indian Institute of Technology, Madras. He has a distinguished academic career and has degrees in Electronics, Economics, Law, Business Finance, Public Relations, Communications, Industrial Relations, Computer Science and Journalism. He has authored and edited 70 books on Computer Science and Economics, which include Petrodollar and the World Economy, C Aptitude, Mastering C, Microprocessor Programming, Mastering C++ and Digital Circuits and Systems etc.. He has filed 100 Patents. During his three decades of service at UVCE he has over 500 research papers to his credit. His research interests include Computer Networks, Wireless Sensor Networks, Parallel and Distributed Systems, Digital Signal Processing and Data Mining.